PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30, 159/00 — A1

(11) International Publication Number: **WO 99/49403**

(43) International Publication Date: 30 September 1999 (30.09.99)

(21) International Application Number: PCT/US99/06575

(22) International Filing Date: 25 March 1999 (25.03.99)

(30) Priority Data:
60/079,469 26 March 1998 (26.03.98) US

(71) Applicant: INCYTE PHARMACEUTICALS, INC. [US/US]; 3174 Porter Drive, Palo Alto, CA 94304 (US).

(72) Inventors: LINCOLN, Stephen, E.; 725 Sapphire Street, Redwood City, CA 94061 (US). HODGSON, David, M.; 567 Addison Avenue, Palo Alto, CA 94301 (US). SPIRO, Peter, A.; 3776 Redwood Circle, Palo Alto, CA 94306 (US). RUSSO, Frank, D.; 939 Rosette Court, Sunnyvale, CA 94086 (US). AKERBLOM, Ingrid, E.; 1234 Johnson Street, Redwood City, CA 94061 (US). HILLMAN, Jennifer, L.; 230 Monrow Drive #17, Mountain View, CA 94040 (US). JONES, Anissa, Lee; 1322 17th Avenue, San Francisco, CA 94122 (US). BRATCHER, Shawn, Robert; 550 Ortega Avenue #B321, Mountain View, CA 94040 (US). COHEN, Howard, Jerome; 3272 Cowper Street, Palo Alto, CA 94306 (US). DUFOUR, Gerard; 3174 Porter Drive, Palo Alto, CA 94304 (US). WOOD, Michael, Peter; 710 Wisconsin Street, San Francisco, CA 94107 (US). KOLESZAR, Alexander, George; 8260 Rinconada Court, Newark, CA 94560 (US). BANVILLE, Steven, C.; 365 Monroe Drive, Palo Alto, CA 94306 (US).

(74) Agents: WILLIAMS, Gary, S. et al.; Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036–2711 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHODS FOR ANALYZING BIOMOLECULAR SEQUENCES

(57) Abstract

Polymer sequences are assembled into bins. A first number of bins are populated with polymer sequences. The polymer sequences in each bin are assembled into one or more consensus sequences representative of the polymer sequences of the bin. The consensus sequences of the bins are compared to determine relationships, if any, between the consensus sequences of the bins. The bins are modified based on the relationships between the consensus sequences of the bins. The polymer sequences are reassembled in the modified bins to generate one or more modified consensus sequences for each bin representative of the modified bins. In another aspect of the invention, sequence similarities and dissimilarities are analyzed in a set of polymer sequences. Pairwise alignment data is generated for pairs of the polymer sequences. The pairwise alignment data defines regions of similarity between the pairs of polymer sequences with boundaries. Additional boundaries in particular polymer sequences are determined by applying at least one boundary from at least one pairwise alignment for one pair of polymer sequences to at least one other pairwise alignment for another pair of polymer sequences including one of the particular polymer sequences. Additional regions of similarity are generated based on the boundaries.

BNSDOCID: <WO___9949403A1_I_>

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

BNSDOCID: <WO___9949403A1_I_>

SYSTEM AND METHODS FOR ANALYZING BIOMOLECULAR SEQUENCES

The present application claims priority to United States Provisional Patent Application Serial No. 60/079,469, entitled Database and System for Storing, Comparing and Displaying Related Biomolecular Sequence Information, filed March 26, 1998 which is incorporated by reference herein for all purposes.

The present invention relates generally to bioinformatics, and particularly to a system and method for analyzing biomolecular sequences.

BACKGROUND OF THE INVENTION

Informatics is the study and application of computer and statistical techniques to the management of information. In genome projects, bioinformatics includes the development of methods to search databases quickly, to analyze nucleic acid sequence information, and to predict protein sequence and structure from DNA sequence data. Increasingly, molecular biology is shifting from the laboratory bench to the computer desktop. Advanced quantitative analyses, database comparisons, and computational algorithms are needed to explore the relationships between sequence and phenotype.

As shown in Fig. 1, a gene 30 is the basic unit of genetic information which is made up of a set of DNA sequences. A gene 30 is transcribed into an RNA primary transcript. This primary transcript is typically spliced to create a mature mRNA, which is then translated into a polypeptide (protein), which performs some function in the cell. An exon 32 is a coding region of the gene 30, while an intron 34 is a control or non-coding region of the gene 30. The most complete representation of a gene 30 is a genomic DNA sequence completely covering, the coding, control and non-coding regions of a gene 30.

BNSDOCID: <WO___9949403A1_I_>

After a gene 30 is transcribed into mRNA, but before the gene 30 is translated into the protein, the gene 30 is edited by removing the introns, and splicing together the remaining exons. For some genes 30, there are several alternative ways the transcript may be spliced, by the optional inclusion or exclusion of each intron or exon. The various arrangements that result are called splice variants.

In Fig. 1, the exons are labeled as 1, 2, 3 and 4. For example, the same gene 30 may generate different mRNA sequences for healthy and diseased tissue, 42 and 44, respectively. The diseased tissue 42 includes sequences from exons 1, 2 and 4, while the healthy tissue 44 includes sequences from exons 1, 2, 3 and 4.

Fig. 2 further illustrates the relationship of expressed sequence tags (ESTs) 46 to mRNA (mRNA1 and mRNA2) and genomic sequences. To form the splice variants, a gene may be transcribed into multiple copies of mRNA. Each mRNA is transcribed into a different cDNA sequence.

An EST 46 is a sampling of a cDNA sequence. ESTs 46 are partial transcript sequences that may cover different parts of the mRNA(s) of a gene, depending on cloning and sequencing strategy.

Researchers generate enormous amounts of data in their attempt to identify gene sequences. In genomic research, DNA, mRNA, and cDNA molecules are broken into fragments, the nucleotide sequence of the fragments are identified, the sequence data for the fragments are input into a database, and a computer program attempts to electronically re-assemble the sequence fragments. There are two types of assembly processes for this data. For genomic data, the DNA from one or more individuals is broken up, individual portions or sequences of the DNA are identified, and then the sequences are reassembled using computer based methods. Any given fragment of a genomic sequence should be represented at approximately the same level,

BNSDOCID: <WO___9949403A1_I_>

and there is theoretically one correct way to reassemble these fragments into a linear sequence representing the original genomic DNA.

In contrast, for expressed sequence tag (EST) based assembly processes, an experimental batch of cDNA is broken into fragments and the nucleotide sequence of the fragments are identified. Since the input mRNA used to generate the cDNA varies widely in abundance, a given fragment of sequence may be present anywhere from once to several thousand times in the set being reassembled. Moreover, because of splice variation, these fragments cannot even theoretically be reassembled into a single linear sequence for each gene.

Fig. 3 is a flowchart of a typical computer-based assembly process for EST data. In step 52, clusters are generated from the EST data. The clustering process groups ESTs based on the similarity between pairs of sequences (pairwise similarity) that make up the ESTs. For example, a computer program, such as BLAST, receives the EST data from two ESTs and generates a score based on the similarity of the bases making up the ESTs. If the score exceeds a predetermined threshold, the ESTs are grouped into the same cluster.

In step 54, within each cluster, the ESTs are assembled into sequence data. Typically, a single cluster will produce many contiguous sequences. Ideally, for each cluster, the goal is to generate a consensus sequence that represents the entire cluster.

This prior art method has two problems. First, the clustering technique tends to overcluster the ESTs. In other words, the method generates too few clusters with too many ESTs in each cluster. Second, the assembly process generates too many consensus sequences. To solve these problems, one prior art method clusters ESTs and selects a single consensus sequence to represent the cluster. For those clusters with multiple consensus sequences,

BNSDOCID: <WO___9949403A1_I_>

another prior art method designates each consensus sequence as a different gene.

However, as discussed above, the same gene may generate multiple cDNA sequences. Therefore, the prior art methods may designate splice variants as different genes. Because individuals can vary in expression of the same gene over long sequences, there is a need for a clustering method that tolerates differences over long sequences. Conversely, cDNA sequences from different genes may be quite similar. Therefore, the clustering method needs to distinguish consensus sequences from different genes from splice variants of the same gene.

Another problem is that existing clustering techniques tend to generate false positives and therefore overcluster. A false positive is a similarity score that exceeds a predetermined threshold, but, in reality, the ESTs are from different parts of the gene or from different genes. To avoid false positives, stringent thresholds can be set for the similarity scores. Conversely, too high a threshold tends to break apart clusters too much, and therefore undercluster. Therefore, a method is needed that avoids under and overclustering problems.

In addition, new EST data continues to be generated and added to existing databases. Therefore, the method needs to be capable of properly clustering and assembling existing ESTs with incremental additions of new EST data.

After the data is clustered, some clusters may generate multiple consensus sequences. A method of identifying and displaying consensus sequences that are splice variants of the same gene is needed.

BNSDOCID: <WO___9949403A1_I_>

SUMMARY OF THE INVENTION

Polymer sequences are assembled into bins. A first number of bins are populated with polymer sequences. The polymer sequences in each bin are assembled into one or more consensus sequences representative of the polymer sequences of the bin. The consensus sequences of the bins are compared to determine relationships, if any, between the consensus sequences. The bins are modified based on the relationships between the consensus sequences. The polymer sequences are reassembled in the modified bins to generate one or more modified consensus sequences for each bin representative of the modified bins.

In another aspect of the invention, sequence similarities and dissimilarities are analyzed in a set of polymer sequences. Pairwise alignment data is generated for pairs of the polymer sequences. The pairwise alignment data defines regions of similarity between the pairs of polymer sequences with boundaries. Additional boundaries in particular polymer sequences are determined by applying at least one boundary from at least one pairwise alignment for one pair of polymer sequences to at least one other pairwise alignment for another pair of polymer sequences including one of the particular polymer sequences. Additional regions of similarity are generated based on the boundaries.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

Fig. 1 is an example of gene expression.

Fig. 2 depicts the relationship of ESTs to mRNA and genomic sequences.

Fig. 3 is a flowchart of a prior art clustering and assembly process.

Fig. 4A is a diagram of client-server system using the present invention.

BNSDOCID: <WO___9949403A1_I_>

Fig. 4B is a diagram of a computer system with a memory storing exemplary procedures and data of the present invention.

Fig. 5A is an exemplary gene bin with a single consensus sequence and EST data.

Fig. 5B is another exemplary gene bin with multiple consensus sequences and EST data.

Fig. 6 is a flowchart of a method of generating gene bins of the present invention.

Fig. 7A illustrates the population and assembling of ESTs in gene bins.

Fig. 7B illustrates the joining of two exemplary gene bins.

Fig. 7C illustrates the splitting of the gene bin of Fig. 7A.

Fig. 8 is a flowchart of a filter applied prior to the assembly or re-assembly process.

Fig. 9 is a flowchart of a method of mapping persistent bin identifiers when new EST data is added to the database.

Fig. 10 is table used for tracking inheritance of old gene bin identifiers to new bin identifiers used with the method of Fig. 10.

Fig. 11 is a flowchart of an alternate embodiment of populating an initial set of gene bins.

Fig. 12 is a flowchart of a method of identification of cross-species gene links.

Fig. 13 is a flowchart of a method of a general method of determining conserved regions across input sequences.

Fig. 14 is an alternate embodiment of the method of Fig. 13.

Fig. 15 is a diagram of three sequences showing regions of similarity and boundaries.

Fig. 16 is more detailed flowchart of the method of Fig. 13.

Fig. 17 is a detailed flowchart of the method of identifying and determining segments with multiple alignments among the received input sequences of Fig. 16.

Fig. 18 shows data structures used with the method of Fig. 17.

BNSDOCID: <WO___9949403A1_I_>

Fig. 19 is an exemplary display of multiple consensus sequences and segment graph.

Figs. 20A and 20B are a flowchart of a method of displaying consensus sequences and a segment graph for identification of splice variants among the consensus sequences as shown in Fig. 19.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In Fig. 4A, a network system is used to retrieve information stored in the biomolecular expression information processing system of the present invention. The major network system components are:

- at least one client computer 60, 62,
- at least one network server 64,
- a storage device 66 storing a gene bin database 68, and
- a firewall gateway server 70 that connects to the internet 72 to access external databases 74.

Fig. 4A depicts the memories 80, 82 of the client computers 60, 62, respectively. On the client computer system 60, a user executes an operating system 84 such as UNIX and a web browser 86 such as Netscape.

The network server 64 has a UNIX operating system 84, an application software module 88 and a relational database management system (RDBMS) 90 such as Oracle. When a user first accesses the application module 88 via the web browser 86, the application module 88 uploads JAVA classes 92 from the server 64 to the client system 80. The JAVA classes 92 include a similarity boundary finder 94 and a template viewer 96 which will be discussed below. The web browser 86 executes the uploaded JAVA classes 98 which use JAVA objects 100 to provide a graphical user interface 102 to the application module 88 for the user. At startup, a subset of the JAVA objects 100 are loaded with data from the database 68.

BNSDOCID: <WO___9949403A1_I_>

To retrieve data from the gene bin database 68, methods within the JAVA classes 98 on client 80 build a SQL statement based on user defined criteria that is passed to a CGI 104 on the network server 64. The CGI 104 then passes the SQL statement to the RDBMS 90. The RDBMS 90 executes the SQL statement and returns the retrieved data to the CGI 104 which, in turn, passes the data back to the client 80. The JAVA classes 98 populate the JAVA objects 100 with the retrieved data, and the results are displayed on the client computer 80.

In an alternate method of retrieving data from the database, methods within the JAVA classes 98 pass a parameter to the CGI script 104 which builds a SQL statement using a SQL Query Generator 106. The SQL statement is passed to the RDBMS 90.

The gene bin database 68 is stored on storage media in a storage device 66 such as a disk drive. In particular, the gene bin database 68 stores the data in tables 108.

The client systems 80, 82, access public domain resources on the Internet 72 via the firewall gateway server 68. The client systems 80, 82, network server 64 and the firewall gateway server 64 are networked via an intranet 109 using TCP/IP protocol.

One of the client systems 82 generates the data that is loaded in the gene bin database 68. A generate_gene_bin procedure 110 uses the methods of the present invention to process expression data 112 to generate gene bins and a gene bin database 114, which will be described below. After generating the gene bin database 114, the client system 82 copies the database onto one of the storage devices 66 on the network server 64 where the copied gene bin database 66 is made available to all users. In an alternate embodiment, the network server 64 generates the gene bin database 68.

BNSDOCID: <WO___9949403A1_I_>

The graphical user interface 98 allows the user to graphically construct search requests to retrieve data from the tables 108 of the gene bin database 68. The commands of the search request are called queries. As described above, either the JAVA classes or the CGI scripts generate the database queries.

The gene bin database 68 has many tables 108 storing information including gene bins, consensus sequences and ESTs.

In Fig. 4B, an exemplary network server computer system 120 stores exemplary procedures and data of the present invention in a memory 122. The memory 122 includes both semiconductor memory and disk memory. A system bus 124 connects a processor 126, a display 128, a keyboard 130, a mouse 132, a network interface 134 that connects to the intranet, a disk drive 136 and the semiconductor memory 122. The procedures and data can also be stored on the disk drive 66. In the memory 122, the procedures include:

- the operating system 84 such as UNIX;
- the Web Browser 86 such as Netscape; and
- a set of application modules 136.

The set of application modules 136 include the following.

- The Generate Gene Bin Procedure 110 creates the gene bins of the present invention.
- The EST data 112 from both private and public databases includes both the raw and processed EST data.
- A block 1 sequence preparation procedure 138 receives the raw EST data and outputs processed EST data for the gene bin database.
- A populate bins procedure 140 populates an initial set of gene bins.
- A Basic Local Alignment Search Tool (BLAST) 142 detects ungapped subsequences in a database that match a given query sequence. BLAST is commonly used and was written at the National Center for Biotechnology Information (NCBI), using a well-grounded statistical

BNSDOCID: <WO___9949403A1_I_>

theory developed by Karlin and Altschul (1993). Matches are based on high-scoring segment pairs (HSPs). Two sequences may have multiple high-scoring segment pairs that are separated by gaps.

- A "phragment" assembly program (PHRAP) 144 assembles shotgun-DNA sequence data such as the processed EST data.
- A representative EST filter 146 generates a representative set of EST sequences to be processed by PHRAP 144.
- An ID&Remove_Bins procedure 148 is used to exclude a predetermined subset of bins from the joining and splitting process of the present invention.
- Cross_match 150 is a computer program for rapid protein and nucleic acid sequence comparison and database searches based on the Smith-Waterman-Gotoh algorithm developed by Phil Green at the University of Washington. In the present invention, Cross_match was modified to obtain sequence alignment comparison results that are independent of the order in which the input sequences are compared.
- An annotate_bins procedure 152 adds annotation data for certain consensus sequences to the database.
- A compare_bins procedure 154 compares the consensus sequences of the gene bins.
- A join_bins procedure 156 joins gene bins.
- A split_bins procedure 158 splits gene bins.
- A FASTX procedure 160 is a database search tool used to compare nucleotide sequences to a peptide sequence database. The procedure is based on the rapid sequence algorithm described by Lipman and Pearson (1988).
- A map_persistent_bin_id procedure 162 maps bin identifiers between old and new versions of the gene bin database.
- The template viewer procedure 96 displays the consensus sequences of the gene bins with their assembled ESTs.
- The gene bin database 68 is stored in the memory 122.

BNSDOCID: <WO___9949403A1_I_>

- A similarity boundary finder 94 finds similar boundaries and segments across input sequences while accommodating for gaps. The similarity boundary finder 94 identifies, aligns and displays consensus segments among an arbitrarily large number of input sequences.
- The RDBMS 90 is also stored in the memory 122.

The similarity boundary finder 94 includes a set of procedures and data structures. The procedures include:

- An id_similar_regions procedure 166 that identifies shared regions of similarity among different sequences and within a sequence;
- A display_con_sequence procedure 168 that displays the shared regions of similarity among different sequences in a spatially aligned manner; and
- A display_segment_map procedure 170 that displays a segment map of the input sequences.

The data structures include:

- input sequence strings 172;
- Cross_match output 174;
- boundary lists 176;
- equivalent boundary lists 178;
- a directed graph array 180; and
- a topological ordering list.

The above-mentioned data structures will be described below.

In Fig. 5A, an exemplary gene bin 200 has a single consensus sequence 202 that represents assembled EST data 204. The term "gene" or "genes" refers to the partial or complete coding sequence of a gene. Gene bins 200 are sequenced-based clusters which have been grouped together. A gene bin 200 is designed to associate or store all the EST sequences 204 for a particular single gene. An EST 204 belongs to only one gene bin 200. Each

BNSDOCID: <WO___9949403A1_I_>

gene bin 200 is associated with the component sequences 204 for a particular single gene. The PHRAP assembly program is run using the ESTs 204 of the bin 200 to generate at least one consensus sequence 202. The consensus sequence 202 acts as a template for that gene. Each base of the assembled sequence represents the consensus of base calls in the component sequences 204 aligned at that position.

As shown in Fig. 5B, in another gene bin 210, the component sequences 212 generate multiple consensus sequences 214, 216, 218. For those gene bins 210 that generate more than one consensus sequence 214, 216, 218, each consensus sequence 214, 216, 218 acts as a template for the gene associated with the gene bin 210. Gene bins 210 with multiple templates or consensus sequences 214, 216, 218 may denote or represent genes with alternative splicing or significant polymorphism.

The gene bins are implemented in tables of the relational database. Each gene bin has a gene bin identifier, each consensus sequence has a consensus sequence identifer and each EST has an identifer. Tables in the database associate the gene bins with consensus sequences and ESTs using the gene bin, consensus sequence and EST identifiers, respectively. Other tables associate the EST data with consensus sequences using the EST and consensus sequence identifiers.

The component sequences or EST data may come from public and private databases.

Fig. 6 is a flowchart of a method of generating gene bins of the present invention used in the generate_gene_bin 110 procedure of Fig. 4B. The flowchart will be described in general, followed by a detailed discussion of each of the steps.

BNSDOCID: <WO___9949403A1_I_>

In general, in step 222, new raw sequence or EST data is received and processed in a set of block 1 procedures (138, Fig. 4B). Step 224 populates an initial set of gene bins with the EST data using the populate_bin procedure (140, Fig. 4B). In step 226, a filter (146 Fig. 4B) is applied to the ESTs in the gene bins to determine a representative set of ESTs which will be assembled using PHRAP. In an alternate embodiment, the filter is not used. In step 228, within each bin, the PHRAP assembler (144, Fig. 4B) is used to assemble the ESTs in the bin to generate one or more consensus sequences. In step 230, the id_&_remove_bins procedure (148, Fig. 4B) identifies a predetermined set of bins and removes them from further processes. In step 232, a compare_bins procedure (154, Fig. 4B) compares the consensus sequences of the bins to determine relationships, if any, between the consensus sequences of the bins. In step 234, a join_bin procedure (156, Fig. 4B) joins bins based on the relationships of the consensus sequences to generate modified bins.

In step 236, the filter (146, Fig. 4B) is applied to the EST data of the modified bins. In an alternate embodiment, the filter is not used. In step 238, within each modified bin, the PHRAP assembler (144, Fig. 4B) is used to re-assemble the ESTs in the modified bins to generate one or more consensus sequences. In step 240, the consensus sequences in the modified bins are compared to determine relationships, if any, between the consensus sequences. In step 242, the modified bins are split based on the relationships of the consensus sequences using the split bin procedure (158, Fig. 4B). In step 244, the method determines whether the comparing, joining and splitting process should repeat. If so, the process continues at step 232. If not, in step 246, bins may be joined based on clone information. In step 248, the filter (146, Fig. 4B) is applied to the EST data of the modified bins. In an alternate embodiment, the filter is not used. In step 250, within each modified bin, the PHRAP assembler (144, Fig. 4B) is used to re-assemble the ESTs in the modified bins to generate one or more consensus sequences. In step 252, the bins are annotated. In step 254, the template viewer procedure (96, Fig.

BNSDOCID: <WO___9949403A1_I_>

4B) displays at least one consensus sequence of a bin spatially aligned with the assembled EST sequences.

In this way, by iteratively comparing consensus sequences and modifying the bins based on the consensus sequences, the method of the present invention provides a set of gene bins that avoids the overclustering and underclustering of the prior art and that tends to group splice variants of the same gene.

Next, each step of Fig. 6 will be described in detail.

Block 1 Sequence Preparation

In step 222, block 1 sequence preparation is performed. After raw sequence data is extracted from a sequencing chromatogram, the raw sequence data passes through a series of filters. First, low quality sequences and those with sequencing artifacts are clipped on the basis of quality scores. Next, recognized 5' and 3' vector sequences are clipped using a method based on dynamic programming. Then regular expression matching to 3' PolyA (or 5' PolyT) patterns is used to clip the mRNA tail.

Next, a series of BLAST comparisons is performed to further filter the sequence data. Low-information segments, such as dinucleotide repeats, are masked – replaced by "n"s--to prevent subsequent spurious matches when the BLAST similarity score is greater than or equal to 150. The "n"s are different from "N"s which are used to represent ambiguities found during sequencing. Raw sequences containing recognized contamination sequences are removed from further bioanalysis when the BLAST similarity score is greater than or equal to 130. Dispersed repetitive elements, such as *Alu*, LINE and MIR are masked when the BLAST similarity score is greater than or equal to 150. Known repetitive elements are present in many copies in the genome. Their functional relevance is very low and they would cause assembly problems if included. Finally, recognized mitochondrial and

BNSDOCID: <WO___9949403A1_I_>

ribosomal RNA sequences are removed based on a BLAST similarity score greater than or equal to 150.

After editing in Block 1, in step 224, the initial bin set is populated with clusters of those sequences having at least fifty bases.

Filter

The filtering steps 226, 236 and 248 will be described below with reference to Fig. 8.

Assembly

In step 228, the PHRAP assembly program generates at least one consensus sequence for each gene bin. The version of PHRAP used in this method was modified to interpret a set of private sequence identifier conventions. In alternate embodiments, other assembly programs, such as FAKII that was developed by Eugene W. Myers, are used. When all bins have at least one consensus sequence, another procedure, such as Cross_match 150 (Fig. 4B), compares all unassigned ESTs to all consensus sequences using a Smith-Waterman based algorithm. An unassigned EST sequence is added to the bin with the consensus sequence that yields the highest Smith-Waterman score. New bins are created for the non-matching unassigned EST sequences.

PHRAP has the advantage of being able to incorporate base quality values into the assembly process. This extra data is essential to achieve the sensitivity and accuracy required for EST assembly.

Comparing Consensus Sequences

In step 232, the bins are modified based on the relationship between the consensus sequences among all the bins. All consensus sequences in all bins are compared to each other using BLAST2. A high BLAST2 score indicates high sequence overlap and identity.

BNSDOCID: <WO___9949403A1_I_>

In an alternate embodiment, in a prescreening operation, all consensus sequences in all bins are compared to each other using BLAST. If the BLAST score exceeds 150 for a pair of consensus sequences, Cross_match is executed using that pair of consensus sequences to verify the BLAST score and generate the local identity.

In another alternate embodiment, instead of using Cross_match to verify the BLAST score, the Smith-Waterman algorithm is used to generate the local identity.

Joining Bins

In step 234, the bins are joined when at least one consensus sequence overlaps a consensus sequence in another bin with at least 82% local identity according to BLAST2. In an alternate embodiment, bins are joined when the local identity is at least 92%. In another alternate embodiment, bins are joined when the local identity is at least 85%.

Re-Assembly

In steps 238 and 250, the PHRAP assembly program generates at least one consensus sequence for each modified gene bin.

Re-Comparing Consensus Sequences for Splitting

In step 240, for those bins having more than one consensus sequence, Cross_match is used to compare the consensus sequences of the re-assembled bins.

In an alternate embodiment, the Smith-Waterman algorithm is used instead of Cross_match.

Splitting Bins

In step 242, using the Cross_match score, bins are split when the overlap between the consensus sequences results in less than 95% identity or the

BNSDOCID: <WO___9949403A1_I_>

length of the alignment is less than fifty base pairs. The consensus sequences with insufficient overlap or alignment are split out to form a new bin.

In step 244, the process of comparing all consensus sequences across all bins, joining bins, re-assembling bins, re-comparing bins and splitting bins repeats until convergence of the database is achieved. Convergence of the database is achieved when the bin compositions do not change significantly between iterations.

In an alternate embodiment, the process of comparing all consensus sequences across all bins, joining bins, re-assembling bins, re-comparing bins and splitting bins repeats for a predetermined number of iterations.

Clone Joining

A single EST clone may be used multiple times to perform sequencing reactions in the laboratory. Therefore, a clone may be associated with multiple sequences. For example, a single clone may be associated with a 5' first-pass sequence, a 5' long-read sequence and a 3' first-pass sequence.

In step 246, after a number of iterations of joining and splitting bins based on their consensus sequences, bins are joined based on clone information. If the 5' sequence of one clone is present in one bin and the 3' sequence from the same clone is present in a different bin, it is likely that the two bins actually belong together in a single bin. Since it is possible that a single clone may be chimeric, bins are joined in this step if there are at least two different clones with a 5' and 3' sequence in each of the bins to be joined.

Bins are not joined if the resulting bin would be very large, having 5,000 or more ESTs. In addition, clone joining is not applied to bins with annotation hits to common genes, nor is clone joining performed on inert bins.

BNSDOCID: <WO___9949403A1_I_>

Annotation

In step 252, using BLAST2 and FASTX, each consensus sequence is compared to the sequences in the GenBank database, one of the external databases available on the internet. Exact hits are annotated and homologs are recorded in the gene bin database. If no match is found for the gene's consensus sequence, the gene is identified as unique in the gene bin database.

Gbpri and gbpept are divisions of the GenBank database. Using BLAST2 searches, hits are collected against the gbpri database. Exact hits are annotated and recorded when the percent identity is greater than or equal to 95% with an alignment length of at least 200 base pairs, to a percent identity greater than or equal to 100% with an alignment length of at least 100 base pairs as summarized below:

percent identity $\geq$ 95%	alignment length $\geq$ 200 base pairs,
percent identity $\geq$ 96%	alignment length $\geq$ 180 base pairs,
percent identity $\geq$ 97%	alignment length $\geq$ 160 base pairs,
percent identity $\geq$ 98%	alignment length $\geq$ 140 base pairs,
percent identity $\geq$ 99%	alignment length $\geq$ 120 base pairs, and
percent identity $\geq$ 100%	alignment length $\geq$ 100 base pairs.

Homologs are recorded when hits have an expectation value (E-value) less than or equal to 1×10^{-8}. The expectation value indicates the expected number of times that an alignment between two sequences might occur by chance. An E-value of zero indicates an exact match while an E-value of one indicates no significant matches were found.

Using BLAST2 searches, hits are collected against the gbpri database. A sequence is annotated as an exact match when the percent identity is equal to 100% with an alignment length of at least 50 base pairs, and both the portion of the template before the match is less than or equal to ten base pairs

BNSDOCID: <WO___9949403A1_I_>

and the portion of the template after the match is less than or equal to ten base pairs.

Using FASTX, hits are collected against the genpept database. The result of the FASTX comparison generates an E-value. Sequences are annotated and assigned homolog status when the E-value is less than or equal to 1×10^{-8}.

Inert Bins

Inert gene bins form a small subset of bins that are not subject to the iterative re-assembly process of steps 232-244. Step 230 identifies and removes the inert bins from the assembly process. Inert bins are very deep, typically having more than 2,000 EST sequences. The inventors found that re-assembly of the inert bins does not significantly affect the existing assembled consensus sequences. Therefore, for the inert bins, new EST sequences that are assigned to the inert bins are aligned to the existing consensus sequences, but the new EST sequences are not used to generate the consensus sequences in the assembly process.

The inert bins are predetermined and are typically well-known and well-characterized genes such as actin or EF-1a.

Populating an Initial Set of Bins with Incremental EST data

In step 224, an initial set of bins is updated with new EST data using the following procedures: assign sequences to bins based on a BLAST comparison, confirm matches and append the EST sequences to the bins for future assembly.

In particular, after filtering using the Block 1 process, the new sequences are assigned to a bin based on the BLAST comparison between the new EST sequences and the current set of consensus sequences. Significant matches are confirmed using Cross_match, a Smith-Waterman based tool that also incorporates base-call confidence scores into the alignment process. Each

BNSDOCID: <WO___9949403A1_I_>

new sequence is added to the bin to which it matches with the highest score. Non-matching sequences create new bins.

Displaying Bins

In step 254, the template viewer procedure displays a bin with at least one consensus sequences with its assembled ESTs. The consensus sequence is displayed at the top of the display, and the ESTs are displayed, starting at the leftmost EST in left-to-right order, below the consensus sequence with one EST to a row.

In Fig. 7A, exemplary ESTs 272 are placed into a bin 274 and assembled to generate a bin 272 with two consensus sequences 276, 278. In Fig. 7B, two exemplary bins 282 and 284 are joined and the ESTs are associated with a single bin 286. In Fig. 7C, the assembled bin 274 of Fig. 7A is split into two bins 292 and 294.

Filter

In Fig. 8, a flowchart of the optional filtering procedure 146 (Fig. 4B) of steps 226 and 236 is shown. The PHRAP assembly program either fails or takes a very long time to execute when the ESTs of a bin have a large local depth. The local depth refers to, for a particular location in the eventual assembly, the number of ESTs whose alignments span that location. To improve the operation and speed of the assembly process, the filter generates a set of representative ESTs for that gene bin that are input to the PHRAP assembler. Since local depth is the problem, the filter effectively removes ESTs located in the regions of greatest local depth, while retaining those ESTs with low local depth. Since some bins may have a very large number of EST sequences, for example, 30,000 or more, the filter reduces the number of ESTs used in the assembly process and thereby speeds up the operation of the assembly process.

BNSDOCID: <WO___9949403A1_I_>

In step 302, for each gene bin starting at the first gene bin, a set of ESTs is initialized. The set of steps in block 304 are then performed for each gene bin. In step 306, a redundancy score is calculated for ESTs in the gene bin. To generate the redundancy score, Cross-match is run on the set of ESTs to obtain the pairwise alignments of the ESTs. Based on the pairwise alignments, the redundancy score for an EST is equal to the minimum, over all the bases of the EST, of the number of matches each base has with respect to the other ESTs in the gene bin. In step 308, the EST with the highest redundancy score is identified. In step 308, the identified EST is removed from the representative set of ESTs. If multiple ESTs have the highest redundancy score, then identify the minimum local depth of the ESTs with the highest redundancy score, and remove the EST with the fewest number of bases having the identified minimum local depth. In this way, the ESTs covering shallow regions tend to remain as representative ESTs, while ESTs in the deeper regions are removed. In addition, ESTs having a shorter sequence length will also tend to be removed.

In step 312, after removing an EST, if the depth of the remaining representative ESTs of the gene bin is greater than a predetermined threshold, the method repeats steps 306 to 310 to determine the next EST to remove. If the depth of the remaining representative ESTs of the gene bin is less than or equal to the predetermined threshold, the process ends for that gene bin.

Cross_match can also incur memory problems and take a long time to execute for bins with large numbers of ESTs. Therefore, in an alternate embodiment, for those bins with large number of ESTs, the ESTs are divided into batches and each batch is processed separately using the method described above for Fig. 8. Prior to assembly, the remaining ESTs are combined into a representative set of ESTs for that bin and are submitted to the assembly process.

BNSDOCID: <WO___9949403A1_I_>

Mapping Persistent Bin Identifiers

Note that a bin identifier can be persistent between database versions. A persistent bin identifier entails the retroactive monitoring of the inheritance of bin identifiers by determining which bins in the newer version of the data base are substantially the same as bins in the older version of the database. Fig. 9 provides a method of mapping persistent bin identifiers using the Map_persistent_bin_id procedure 162 (Fig. 4B). In this method, bin identifiers are mapped from an old set of bins of an old database to a new set of bins of a new database. The method is independent of the process used to generate the bins. Using this method, there is no need to track a bin identifier through the many steps of processing of Fig. 6 or to generate and compress a processing history into a compact interpretable form.

In step 322, for all pairs of bins, each pair having one bin from the old database and one bin from the new database, a two-sided score that includes a forward score and a reverse score is determined as follows:

$$\text{Forward Score} = \frac{\#\text{ ESTs in common between the old and new bin in the pair of bins}}{\text{total \# of inherited ESTs in the new bin from the old set of bins}}$$

$$\text{Reverse Score} = \frac{\#\text{ ESTs in common between the old and new bin in the pair of bins}}{\text{total \# of inheritable ESTs in the old bin}}$$

For each pair of bins, both the forward score and the reverse score have the same numerator. The denominator of the forward score is the total number of inherited ESTs in the new bin. In other words, the total number of ESTs in the new set of bins that were present in the old set of bins. The denominator of the reverse score is the total number of ESTs in the old bins.

BNSDOCID: <WO___9949403A1_I_>

In step 324, for each new bin, all Reverse Scores greater than or equal to a predetermined reverse score threshold, such as 90%, are identified in order to identify a subset of potentially inheritable bin identifiers, and all Forward Scores are ranked.

In step 326, for each new bin, the new bin identifier is mapped to the old bin identifier in the subset of potentially inheritable bin identifiers that has the highest Forward Score. In Fig. 10, a table 328 in the database store the mapping of old bin identifiers to new bin identifiers.

Alternate Method of Populating an Initial Set of Gene Bins

Fig. 11 is a flowchart of an alternate embodiment of populating the initial set of gene bins of step 224 of Fig. 6. In step 332, each EST sequence is placed in its own bin so that each EST is a consensus sequence. In step 334, the consensus sequences of the bins are compared to determine relationships, if any between the consensus sequences of the bins. Step 334 of Fig. 11 is the same as step 232 of Fig. 6. In step 336, the bins are joined based on the relationships of the consensus sequences. Step 336 of Fig. 11 is the same as step 234 of Fig. 6.

Cross-Species Gene Links

Sets of gene bins can be assembled not only for human sequence data, but also for other organisms. In these gene bins, the same gene may appear across multiple species. Genes sufficiently common to be captured by the libraries for a given species that are grouped together by the assembly process will appear in the database represented at the sequence level by one or more consensus sequences from one or more gene bins.

To associate bins from multiple organisms to show that they represent the same gene, in step 338 of Fig. 12, consensus sequences from assembled bin sets from each species are compared using BLAST. In step 340, for those comparison results exceeding a predetermined threshold, a first species

BNSDOCID: <WO___9949403A1_I_>

identifier, the first species gene bin identifier, the first species consensus sequence identifier and a second species identifier with its second species gene bin identifier, a second species consensus sequence identifier are stored in a table in the database to provide a cross-reference of common genes among species.

Similarity Boundary Finder

The purpose of the similarity boundary finder is to identify and then extract information about regions of similarity between input sequences, as well as unique regions. Regions of similarity are patterns that occur at least once in two or more input sequences, or that occur at least twice in a single input sequence. A segment is a region of similarity, or is designated as such, when the difference between patterns from different input sequences is deemed as biologically unimportant. Input sequences have at least one and typically many segments.

In Fig. 13, a flowchart of a general method of determining conserved regions across input sequences 174 (Fig. 4B) used by the similarity boundary finder 94 (Fig. 4B) is shown. In step 352, the initial pairwise alignment criteria is set. Since the pairwise alignment tool is Cross_match, the criteria includes a minimum length and a score threshold at which a homologous sequence or region of similarity is identified. In step 354, pairwise alignment data 176 (Fig. 4B) is generated for all pairs of input sequences using Cross_match. In step 356, based on the pairwise alignment data, boundaries of aligned sequence portions are identified. All boundaries of all aligned sequence portions are determined by iteratively applying all identified boundaries to previously identified aligned sequence portions. In step 358, an average number of boundaries per input sequence is determined. In step 360, if the average number of boundaries is greater than or equal to a predetermined threshold, the process proceeds to step 362. In step 362, the pairwise alignment criteria is modified to increase the requirements for pairwise alignment such that the

BNSDOCID: <WO___9949403A1_I_>

number of aligned sequence portions will be reduced and the process repeats at step 354. If the average is less than the predetermined threshold, step 364 displays the input sequences with their aligned sequence portions and boundaries. In one embodiment, a user sets the predetermined threshold number of sequences to be compared to the average.

Fig. 14 is an alternate embodiment of the general method of the similarity boundary finder of Fig. 13. Fig. 14 is different from Fig. 13 because the pairwise alignment data is generated only once. As in Fig. 13, in step 352, the pairwise alignment criteria are set; and, in step 354, the pairwise alignment data for pairs of input sequences are generated. At this point the alternate embodiment of Fig. 14 differs from that shown in Fig. 13. In step 365, the pairwise alignment data are ordered according to the likelihood of generating short segments. A pairwise alignment is considered likely to yield short segments according to the extent to which the aligned regions of the sequences involved are also contained in other pairwise alignments. In addition, the likelihood is considered especially high if there is another pairwise alignment involving the same two sequences and containing the majority or the entire extents of the aligned regions.

In step 367, based on the ordered pairwise alignment data contained in the pairwise alignment data processed so far, the boundaries of aligned sequence portions are identified, and all boundaries of all shared sequence portions are determined by iteratively applying all identified boundaries to aligned sequence portions.

In step 368, the average distance between boundaries in the input sequences is determined. In step 369, if the average is greater than or equal to a predetermined threshold and if there are more pairwise alignments to process, the process proceeds to step 370 to get the next pairwise alignment and the process repeats at step 367. If the average distance between boundaries is less than the predetermined threshold and if there are no more pairwise

BNSDOCID: <WO___9949403A1_I_>

alignments to process, in step 364, the input sequences are displayed with their boundaries.

Step 364 is the same for Fig. 13 and Fig. 14. To display the input sequences with their boundaries, depending on the embodiment, the id_similar_regions procedure 166 of Fig. 4B implements either steps 352-362 of Fig. 13 or steps 352, 354, 365-370 of Fig. 13. The display_con_sequence procedure 168 and the display_segment_map procedure 170 of Fig. 4B implements step 364 of Figs. 13 and 14.

In Fig. 15, three exemplary sequences are shown - Sequence 1, Sequence 2 and Sequence 3. Sequences 1 and 2 have a first region of similarity with boundaries Boundary 1 and Boundary 2. Sequences 2 and 3 have a second region of similarity with boundaries Boundary 3 and Boundary 4. Since Boundary 3 falls in the middle of the first region of similarity, the present invention will apply Boundary 3 to Sequence 1 thereby splitting the first region of similarity into two portions. Since Boundary 2 falls in the middle of the second region of similarity, Boundary 2 is applied to Sequence 3 to split the second region of similarity into two portions.

Figs. 16A and 16B are a more detailed flowchart of the method of Fig. 13. In step 372, input sequences are received. Preferably the input sequences are consensus sequences of EST assemblies. Alternately other sequences can be received such as genomic sequence data. Auxiliary data may also be received with the input sequences such as assembly depth, base call quality scores, and tissue or disease-state categorization. In step 374, as described above, the initial pairwise alignment criteria is set. In step 376, pairwise alignments between the input sequences are identified. In addition, pairwise alignments between the input sequences and their reverse complements are identified. In step 378, for each pairwise alignment, the boundaries of the alignment in each sequence, the locations of all insertions and deletions in the alignments and the orientation of each sequence are identified. In step 380,

BNSDOCID: <WO___9949403A1_I_>

the pairwise alignments are split at large gaps. A large gap is a gap that exceeds a predetermined threshold gap length in the pairwise alignments. A user can set the predetermined gap length. For each large gap, the pairwise alignment is subdivided at the large gap to form two new shorter pairwise alignments. The ends of the gap are boundaries. In step 382, any sequences whose alignments are primarily to their reverse complements are replaced with their reverse complements. This step is performed to simplify the display. In step 384, based on the pairwise alignment data, the boundaries of aligned sequence portions are identified. All boundaries of all regions of similarity between sequences are determined by iteratively applying all identified boundaries to all aligned sequence portions. Steps 358, 360 and 362 are the same as described above and the description will not be repeated.

After step 360, in step 390, based on the pairwise alignment data and the boundaries, segment instances are identified. A segment instance is a region of a sequence between a pair of adjacent similarity boundaries. In step 392, similar segment instances (e.g., from different input sequences) are clustered into segment groups.

In step 394, the segment instances are multiply aligned into segment groups. In one implementation, the alignment along a tree method is used, except that instead of using profiles as guides in aligning two multiple alignments, the gapping that is specified by one of the generated pairwise alignments that has a segment from each multiple alignment is used. The structure of the tree is determined by an ordering of the sequence pairwise alignments. Segment instances are iteratively clustered into binary trees by merging, for each pairwise alignment, the pair of trees containing the two segment instances contained in the alignment. The pairwise alignments are processed in increasing order of the sum of the lengths of the two aligned regions because such an ordering appears likely to join more similar segments before more dissimilar segments. However, other orderings can be used. A pairwise alignment is ignored if its two aligned segment instances are already in the

BNSDOCID: <WO___9949403A1_I_>

same tree. Although the multiple alignment yielded by this method may not be optimal, this method is fast because it does not require calculation of new pairwise alignments.

In step 396, the consensus segment for each of the segment groups is determined by selecting, for each position in the multiple alignment, the base call having the highest quality score from among the base calls at the corresponding positions in the segment instances. A gap quality score is assigned to equal the average score of the two bases on either side of the gap. Ties are resolved by selecting the base call occurring in the largest number of segment instances at the highest quality score. If there is still a tie, an unambiguous base call is chosen instead of a gap, and a gap is chosen over an ambiguous base call. If there is still a tie among unambiguous base calls, assign an "N" to that position in the consensus segment. For each position in the consensus sequence, the quality score is defined as the highest score among the segment instances at that position. The assembly depth and tissue counts are the sums of the equivalent quantities for the segment instances.

In step 398, junctions between segment groups are identified. A junction occurs when two segment instances, one from each group, are adjacent in any sequence. In step 400, for nucleotide input sequences and their consensus sequences, likely splice junction sequences are identified. In step 402, the input sequences are displayed with their boundaries.

Fig. 17 is a detailed flowchart of the method of identifying and determining segments with multiple alignments among the received input sequences of step 386 of Figs. 16A and 16B. In step 422, for each sequence, a boundary list 178 (Fig. 4B) is created and populated with the sequence's left and right endpoints. In step 424, the left and right endpoints of all pairwise alignments involving the sequence is added to that sequence's boundary list. An equivalent boundary list 180 (Fig. 4B) associates the equivalent boundaries of

BNSDOCID: <WO___9949403A1_I_>

the pairwise alignments among the input sequences. In step 426, a queue of boundaries to be processed is generated. Initially, the queue includes all of the above sequence and alignment endpoints. The queue may also be implemented as another list. In step 428, for each boundary in the queue, a spanning list of all pairwise alignments spanning the boundary location in a corresponding sequence is created. In step 430, for each pairwise alignment in the spanning list, the pairwise alignment is subdivided by adding the boundary to the boundary list of the input sequence associated with the pairwise alignment if the boundary list does not already contain a boundary at this location, and this boundary is added to the queue for processing.

Fig. 18 shows data structures used with the method of Fig. 17 that reflect the exemplary sequences, alignment and boundaries of Fig. 15. Initially, each sequence has a boundary list with its starting point, S1, S2 and S3, and end point, E1, E2 and E3, respectively. Each initial boundary list also has boundaries from the pairwise alignment data. In Fig. 18, the boundaries are uniquely designated as "Bx" where x refers to a boundary number. Boundaries B1 and B2 of sequences 1 and 2 are aligned. In practice, boundary B1 will most likely occur at a different location in sequence 1, such as fifty, from boundary B1 in sequence 2, such as seventy. However, for simplicity, both boundaries are designated as B1. Boundaries B3 and B4 of sequences 2 and 3 are also aligned. Referring also to Fig. 15, boundaries 1, 2, 3 and 4 of Fig. 15 are the same as B1, B2, B3 and B4 of Fig. 18.

In Fig. 18, another data structure, such as a list, is used to associate the equivalent boundaries among the sequences, such as B1 from sequence 1 and B1 from sequence 2.

The boundary lists for each sequence are shown after applying the method of Figs. 13, 14 and 17 described above. Note that boundary B3 is added to the list for Sequence one and boundary B2 is added to the list for Sequence three.

BNSDOCID: <WO___9949403A1_I_>

Other lists, such as the list of the pairwise alignments spanning the boundary locations, are also used.

In Fig. 19, the input sequences and their segments are displayed. An exemplary display 440 has an upper portion 442 displaying the input sequences AA, BB.c, and CC with aligned consensus segments 443. One input sequence with all its consensus segments is displayed horizontally on a single line. For simplicity, the segments are numbered. In practice, each similar segment has a unique color. Input sequence BB.c is the reverse complement as indicated by the ".c" extension.

The rows of input sequences are displayed in an order that positions more similar sequence pairs closer together than less similar pairs based on the number of similar base pairs of each input sequence.

The lines 444 between segments indicate junctions. The junctions are drawn at the endpoints at which the segments meet. An alignment between a region of a sequence and its reverse complement is displayed with an "X" pattern.

In a lower display 444, a segment graph shows the relationship among the aligned segments. The segments are numbered one through fourteen and each segment is shown once. Again, the lines indicate junctions between segments. Note that segment 6 is a likely alternatively spliced exon because input sequence AA includes segment 6 while input sequence BB.c does not include segment 6 as indicated by the curved line connecting segments 5 and 7. The segments of the segment graph are also vertically aligned with respect to the segments of the input sequences in the upper display. Segments 8 and 9 are repeating sequences. The method of the present invention results in repeating sequences being identified both within a single input sequence and among two or more input sequences.

BNSDOCID: <WO___9949403A1_I_>

In a preferred embodiment, the input sequences are consensus sequences from the gene bins.

Figs. 20A and 20B are a flowchart of a method of displaying input sequences and the segment graph of Fig. 19 for identification of splice variants among the input sequences. In step 452, the input or consensus sequences and their segments are received. In step 454, the relative horizontal ordering of segments in the display is determined by clustering segment instances within segment groups into subsets that will share the same horizontal location. In step 456, the relative horizontal ordering of segment instances is represented using an acyclic directed graph 182 (Fig. 4B). The vertices of the acyclic directed graph represent segment subsets and the edges indicate the horizontal adjacency of the segment subsets, with the edge direction dictated by the two segment subsets' left-right ordering. The acyclic directed graph is initialized as a set of unconnected directed paths, each path representing the ordering of segment instances within one input sequence.

In step 458, a list of all pairs of similar segment instances is created and the list is sorted. The list is sorted, first in descending order of the lengths of each pair's input sequences, then by whether the pair has the same orientation, then in ascending order of the two segment instances' average location within their respective input sequences.

In step 460, for each segment instance pair in the sorted list, starting from the beginning of the list, attempt to merge the subsets to which the two segment instances belong, if the segment instances in the pair belong to different subsets of segments. In other words, when a merge is to be performed, identify the two vertices in the acyclic directed graph corresponding to the two subsets, and merge the subsets only if doing so will not cause a cycle to be added to the acyclic directed graph when the corresponding graph vertices are merged.

BNSDOCID: <WO___9949403A1_I_>

In step 462, the absolute positions of segment subsets in the display are determined by:

(a) creating a topological ordering of all segment subsets 184 (Fig. 4B), i.e., a list of the subsets ordered left-to-right in a manner consistent with their individual relative orderings;

(b) creating directed trees of connected segment subsets, each tree extending leftward from its root;

(c) removing the leftmost segment subset from the topological ordering to form the root of a new directed tree:

(i) from left-to-right in the topological ordering, for each segment subset, if its left end has any junctions to the right end of any segment subset already in the tree, remove it from the topological ordering and add it to the new tree, making it a child node of the left subset with the rightmost right end, and position its left end at a specified minimum separation distance to the right of its parent's right end, and

(ii) for every previously created tree, if there are any junctions between the right end of any segment subset in this tree and the left end of any segment subset in the previous tree, position this tree relative to the previous tree so that the segment subsets involved in all such junctions are separated by at least the minimum separation distance, and so that the segment subsets involved in at least one such junction are separated by exactly the minimum separation distance; and

(d) Until no subsets remain in the topological ordering, remove the leftmost subset remaining in the topological ordering to form the root of another new directed tree, and repeat steps (c)(i) and (c)(ii) for this new directed tree.

The relative positioning of the above trees defines one or more clusters of connected segment subsets. The segment subsets within each cluster form a connected graph via their junctions and segment subsets in different clusters have no left-to-right junctions to each other. All such clusters are aligned so that the left end of the leftmost segment subset in each cluster is located at position zero.

BNSDOCID: <WO___9949403A1_I_>

In step 464, the input sequences are ordered vertically by:

creating an ordering of all pairs of input sequences, sorted in decreasing order of the total lengths of all pairwise alignments between each input sequence pair;

creating lists of vertically ordered input sequences, by processing, in order, pairs of input sequences as follows:

starting with each sequence being in its own one-sequence list, then in the ordering created in the previous step, if two input sequences in a pair belong to different lists, append one list to the other; and

if, at the end, there are two or more lists, arrange the lists vertically in decreasing order of their numbers of consensus sequences.

In an alternate embodiment, for multiple lists, the topmost list to display will be determined based on the length of the input sequences.

In step 466, the vertical (row) positions of consensus segments in the segment graph are determined by:

sorting all segment instances in decreasing order of the length of the corresponding sequence;

starting with a segment graph having only empty rows, for each segment instance in the sorted list, if the corresponding segment subset does not yet have a position in the graph, add the corresponding consensus segment to the topmost row of the graph where the consensus segment can be positioned at the horizontal location of the segment subset and be at least the minimum separation distance from all other consensus segments already positioned the row.

In an alternate embodiment, a consensus segment is added to the topmost row in which it fits and which contains the consensus segment of a second segment subset with which the first segment subset shares a left-to-right junction. If there is no such row, then the consensus segment is added to the topmost row in which it fits.

BNSDOCID: <WO___9949403A1_I_>

In this way, the similarity boundary finder processes the output of the pairwise alignment to reliably identify conserved regions in a manner consistent with all of the pairwise alignment data, no matter how complex. Therefore, the similarity boundary finder can be used to aid in determining alternative splicing of gene by displaying putative variants, that is, segments which may correspond to putative alternatively spliced exons or groups of exons.

The input sequences to the similarity boundary finder are not limited to consensus sequences of the gene bins. The similarity boundary finder can be used to determine genomic to cDNA alignments by processing the genomic and cDNA sequence data as the input or consensus sequences described above. The similarity boundary finder can also be used to identify similar regions of homologous sequences including cross-species homologs by processing sequence data from two different species as the input or consensus sequences described above.

In addition, the similarity boundary finder can be used to determine sequence polymorphisms, such as single nucleotide polymorphisms (SNPs) - including substitutions, insertions and deletions. This can be done by disallowing substitutions in the Cross_match pairwise alignments by setting the magnitude of the mismatch penalty greater than twice that of the gap initiation penalty to force SNPs to appear as gaps in the alignments, and by setting the minimum gap length to zero within a segment, to force SNPs to form individual single base segments.

The similarity boundary finder can also used to determine tissue differentiation among the segments in the consensus sequences. The similar and dissimilar segments are correlated with a tissue category to form subsets having a common tissue category. Each subset may include both similar and dissimilar segments. The polymer sequences are displayed as shown in Fig. 19. Each subset of segments is displayed with a unique color such that the colors of the segments indicate regions where expression is specific to a single tissue category.

BNSDOCID: <WO___9949403A1_I_>

In an alternate embodiment, instead of correlating and identifying tissue categories, the segments are correlated with a disease state and each disease state is uniquely identified on the display.

In yet another alternate embodiment, the segments are correlated with a developmental stages, and each developmental stage is uniquely identified on the display.

The present invention solves many problems of identifying genes from many heterogeneous sequences. The invention removes chimeric clones, removes construction artifacts, masks repetitive elements, splits close homologs, merges gene bins with apparent splice variation into a single gene bin, and trims low accuracy tails. The present invention also provides a visual display of the consensus sequences of the gene bins for identification of splice variants.

While the present invention has been described with reference to a few specific embodiments, the description is illustrative of the invention and is not to be construed as limiting the invention. Various modifications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

BNSDOCID: <WO___9949403A1_I_>

WHAT IS CLAIMED IS:

1. A method of assembling polymer sequences comprising the steps of:

populating a first number of bins with polymer sequences;

assembling the polymer sequences in each bin into one or more consensus sequences representative of the polymer sequences of the bin;

comparing the consensus sequences of the bins to determine relationships, if any, between the consensus sequences of the bins;

modifying the bins based on the relationships between the consensus sequences of the bins; and

reassembling the polymer sequences in the modified bins to generate one or more modified consensus sequences for each bin representative of the modified bins.

2. The method of claim 1 wherein said step of modifying includes:

joining two of the bins when consensus sequences of the two bins meet predefined overlap criteria.

3. The method of claim 1 wherein said step of modifying includes splitting one of the bins into two bins when consensus sequences of the one bin meet predefined distinctiveness criteria.

4. The method of claim 1 further comprising the step of:

repeating said steps of comparing, modifying and reassembling for the modified bins.

5. The method of claim 4 wherein said step of repeating is performed a predetermined number of times.

6. The method of claim 4 wherein each repetition of said steps of comparing, modifying and reassembling is denoted as an iteration, wherein the modified bins form a set of modified bins; and

BNSDOCID: <WO___9949403A1_I_>

prior to said step of repeating,

identifying bins that have been already been modified in a prior iteration;

eliminating the identified bins from the set of modified bins.

7. The method of claim 6 wherein said step of repeating is performed until the set of modified bins is empty.

8. The method of claim 1 further comprising the steps of:

prior to said reassembly step, applying a filter to the polymer sequences in each bin to identify a representative subset of polymer sequences, and wherein said step of reassembling reassembles the representative subset of polymer sequences to generate the modified consensus sequences.

9. The method of claim 1 further comprising the steps of:

prior to said assembly step, applying a filter to the polymer sequences in each bin to identify a representative subset of polymer sequences; and wherein said step of assembling assembles the representative subset of polymer sequences to generate the consensus sequences.

10. The method of claim 1 wherein the polymer sequences include expressed sequence tags, and the bins are gene bins representing all or a portion of an expressed gene.

11. The method of claim 10, wherein when at least one gene bin has at least two consensus sequences, the at least two consensus sequences represents a splice variant of at least a portion of the expressed gene.

12. The method of claim 1 further comprising the steps of:

identifying one or more homologs between at least one of the consensus sequences and an external database, and annotating the identified the at least one consensus sequence with an external database identifier.

BNSDOCID: <WO___9949403A1_I_>

13. The method of claim 1 wherein said steps of comparing and modifying include:

identifying a subset of bins having 3' and 5' polymer sequences from a same clone; and

joining the identified subset of bins.

14. A method of assembling polymer sequences comprising the steps of:

populating a first number of bins with polymer sequences;

assembling the polymer sequences in each bin into one or more consensus sequences representative of the polymer sequences of the bin;

joining ones of the bins based on similarity between the one or more consensus sequences of the bins;

splitting ones of the bins into two or more split bins based on dissimilarity between the consensus sequences of the bins;

reassembling the polymer sequences in the modified bins to generate one or more modified consensus sequences for each bin representative of the modified bins; and

repeating said steps of joining, splitting and reassembling using the joined and split bins.

15. The method of claim 14 further comprising the step of:

prior to said step of reassembling, applying a filter to the polymer sequences in each bin to identify a representative set of polymer sequences,

wherein said step of reassembling reassembles the representative set of polymer sequences to generate the modified consensus sequences.

16. The method of claim 14 wherein the polymer sequences include expressed sequence tags, and the bins are gene bins representing at least a portion of an expressed gene.

BNSDOCID: <WO___9949403A1_I_>

17. The method of claim 14, wherein when at least one gene bin has at least two consensus sequences, the at least two consensus sequences represent at least two splice variants of the at least portion of the expressed gene.

18. The method of claim 14 further comprising the steps of:

prior to said assembly step, applying a filter to the polymer sequences in each bin to identify a representative subset of polymer sequences; and wherein the step of assembling assembles the representative subset of polymer sequences to generate the consensus sequences.

19. A computer system for providing biomolecular information comprising:

a processor; and

a memory, coupled to the processor, for storing instructions that:

populate a first number of bins with polymer sequences;

assemble the polymer sequences in each bin into one or more consensus sequences representative of the polymer sequences of the bin;

compare the consensus sequences of the bins to determine relationships, if any, between the consensus sequences of the bins;

modify the bins based on the relationships between the consensus sequences of the bins; and

reassemble the polymer sequences in the modified bins to generate one or more modified consensus sequences for each bin representative of the modified bins.

20. The computer system of claim 19 wherein said instructions that modify include instructions that:

join two of the bins when consensus sequences of the two bins meet predefined overlap criteria.

21. The computer system of claim 19 wherein said instructions that modify include instructions that split one of the bins into two bins when consensus sequences of the one bin meet predefined distinctiveness criteria.

BNSDOCID: <WO___9949403A1_I_>

22. The computer system of claim 19 further comprising instructions that repeat said instructions that compare, modify and reassemble the modified bins.

23. The computer system of claim 22 wherein said instructions that compare, modify and reassemble are repeated a predetermined number of times.

24. The computer system of claim 22 wherein each repetition of said instructions that compare, modify and reassemble is denoted as an iteration, wherein the modified bins form a set of modified bins; and
prior to said instructions that repeat, further including instructions that:
identify bins that have been already been modified in a prior iteration; and
eliminate the identified bins from the set of modified bins.

25. A computer program product for assembling polymer sequences, the computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:
a first set of instructions that populates a first number of bins with polymer sequences;
an assembler that assembles the polymer sequences in the bins into one or more consensus sequences representative of the polymer sequences of the bin;
a second set of instructions that executes the assembler using the first number of populated bins;
a third set of instructions that compares the consensus sequences of the bins to determine relationships, if any, between the consensus sequences of the bins;
a fourth set of instructions that modifies the bins based on the relationships between the consensus sequences of the bins; and
a fifth set of instructions that executes the assembler using the modified bins to generate a new set of consensus sequences for the modified bins.

BNSDOCID: <WO__9949403A1_I_>

26. The computer program product of claim 25 wherein said fourth set of instructions include instructions that:

join two of the bins when consensus sequences of the two bins meet predefined overlap criteria.

27. The computer program product of claim 25 wherein said fourth set of instructions include instructions that split one of the bins into two bins when consensus sequences of the one bin meet predefined distinctiveness criteria.

28. The computer program product of claim 25 further comprising instructions that

repeat said third, fourth and fifth sets of instructions that compare, modify and reassemble the modified bins, respectively.

29. The computer program product of claim 25 wherein said third, fourth and fifth sets of instructions that compare, modify and reassemble, respectively, are repeated a predetermined number of times.

30. A method for analyzing sequence similarities and dissimilarities in a set of polymer sequences, the method comprising the steps of:

generating pairwise alignment data for pairs of the polymer sequences, the pairwise alignment data defining regions of similarity between the pairs of polymer sequences with boundaries;

determining additional boundaries in particular polymer sequences by applying at least one boundary from at least one pairwise alignment for one pair of polymer sequences to at least one other pairwise alignment for another pair of polymer sequences including one of the particular polymer sequences; and

generating additional regions of similarity based on the boundaries.

31. The method of claim 30 wherein the polymer sequences include a first sequence and a second sequence, the generated pairwise alignment data

BNSDOCID: <WO___9949403A1_I_>

between the first and second sequence includes a first region of similarity, the pairwise alignment data including gaps further comprising the step of:

identifying at least two distinct regions of similarity in the first region of similarity from the pairwise alignment data, where intervening sequence portions are distinct.

32. The method of claim 30 wherein the intervening sequence portions are distinct when a number of adjacent polymers in the intervening sequence portions exceeds a predetermined gap threshold.

33. The method of claim 30 wherein the regions of similarity form segments, and dissimilar regions also form segments, and further comprising the step of:

when numbers of segments is greater than or equal to predetermined thresholds, modifying a set of criteria to generate the pairwise alignment data, and repeating said steps of generating the pairwise alignment data, determining all boundaries and generating the additional regions of similarity whereby the total number of segments is reduced.

34. The method of claim 30 wherein the regions of similarity form segments, and dissimilar regions also form segments, and further comprising the step of:

when lengths of segments are greater than or equal to predetermined thresholds, modifying a set of criteria to generate the pairwise alignment data, and repeating said steps of generating the pairwise alignment data, determining all boundaries and generating the additional regions of similarity whereby the total number of segments is reduced.

35. The method of claim 30 further comprising the step of:

displaying the polymer sequences with the regions of similarity being spatially aligned with each other.

36. The method of claim 35 wherein said step of displaying displays sequence differences and similarities in the polymer sequences, and includes the steps of:

BNSDOCID: <WO___9949403A1_I_>

displaying the polymer sequences, the polymer sequences also having regions of dissimilarity, wherein for each polymer sequence the regions of similarity and dissimilarity are horizontally aligned based on the position of each region in the polymer sequence, wherein the regions of similarity among different polymer sequences are vertically aligned.

37. The method of claim 30 wherein said step of generating additional regions of similarity includes the step of:

subdividing the regions of similarity using the boundaries.

38. The method of claim 37, further comprising the step of repeating said step of subdividing.

39. The method of claim 30, further comprising the steps of:

repeating said steps of generating pairwise alignment data, determining additional boundaries and generating additional regions of similarity; and

changing pairwise alignment criteria to generate the pairwise alignment data at each step of repeating.

40. The method of claim 30 further comprising the steps of:

identifying repetitive sequences within a particular sequence.

41. The method of claim 39 further comprising the step of:

displaying the identified repetitive sequences with a unique designation.

42. The method of claim 30, wherein the polymer sequences also have regions of dissimilarity, and further comprising the steps of:

identifying the regions of similarity and dissimilarity; and

correlating the regions of similarity and dissimilarity to a biological property.

43. The method of claim 30, further comprising the steps of:

identifying sequence polymorphisms; and

BNSDOCID: <WO___9949403A1_I_>

correlating the sequence polymorphisms to a biological property.

44. The method of claim 30, further comprising the step of deriving a consensus segment for each region of similarity representative of the polymer sequences of the region of similarity.

45. The method of claim 30 wherein the regions of similarity represent exons.

46. The method of claim 30 wherein the set of polymer sequences includes at least two consensus sequences representing splice variants of at least a portion of a transcribed nucleic acid sequence, and where the regions of similarity include exons and portions of exons.

47. The method of claim 34 wherein the regions of similarity represent exons, and said step of displaying is used to detect splice variants.

48. The method of claim 30 wherein the polymer sequences include at least one genomic sequence and at least one transcribed nucleic acid sequence,and wherein the regions of similarity include exons and portions of exons.

49. The method of claim 30 wherein the polymer sequences are consensus sequences.

50. The method of claim 30 further comprising the step of generating a consensus segment representing common regions of similarity among the polymer sequences, and the regions of dissimilarity being referred to as unique segments, and further comprising the step of:

displaying the consensus segments and the unique segments in a segment graph, the segment graph displaying each consensus segment and each unique segment once based on the position of the consensus segment and the unique segments in the polymer sequences.

BNSDOCID: <WO___9949403A1_I_>

51. The method of claim 41 wherein the polymer sequences include at least two transcribed nucleic acid sequences from different tissue categories, and further comprising the steps of:

correlating the regions of dissimilarity with the tissue categories.

52. The method of claim 41 wherein the polymer sequences include at least two related gene sequences and wherein the regions of similarity include conserved regions between the at least two related gene sequences.

53. The method of claim 30 wherein the polymer sequences include at least two related gene sequences and wherein the regions of similarity include conserved regions between the at least two related gene sequences.

54. The method of claim 30 wherein the polymer sequences include at least two transcribed nucleic acid sequences from different disease states, and further comprising the step of:

correlating the regions of dissimilarity with the disease states.

55. The method of claim 30 wherein the polymer sequences include at least two transcribed nucleic acid sequences from different developmental stages, and further comprising the step of:

correlating the regions of dissimilarity with the developmental stages.

56. A computer system for analyzing sequence similarities and dissimilarities in a set of polymer sequences, comprising:

a processor; and

a memory, coupled to the processor, for storing instructions that:

generate pairwise alignment data for pairs of the polymer sequences, the pairwise alignment data defining regions of similarity between the pairs of polymer sequences with boundaries;

determine additional boundaries in particular polymer sequences by applying at least one boundary from at least one pairwise alignment for one pair of polymer

BNSDOCID: <WO___9949403A1_I_>

sequences to at least one other pairwise alignment for another pair of polymer sequences including one of the particular polymer sequences; and

generate additional regions of similarity based on the boundaries.

57. The computer system of claim 56 wherein the memory further includes instructions that:

display the polymer sequences with the regions of similarity being spatially aligned with each other.

58. The computer system of claim 57 wherein the polymer sequences also have regions of dissimilarity, said instructions that display include instructions that:

horizontally align the regions of similarity and dissimilarity based on the position of each region in the polymer sequence; and

vertically align the regions of similarity among different polymer sequences.

59. The computer system of claim 56 wherein the memory further includes instructions that:

identify repetitive subsequences within a particular sequence.

60. The computer system of claim 56 wherein the polymer sequences also have regions of dissimilarity, and the memory further includes instructions that:

identify the regions of similarity and dissimilarity; and

correlate the regions of similarity and dissimilarity to a biological property.

61. The computer system of claim 56 wherein the memory further includes instructions that:

identify sequence polymorphisms; and

correlate the sequence polymorphisms to a biological property.

62. A computer program product for assembling polymer sequences, the computer program product for use in conjunction with a computer system, the computer program

BNSDOCID: <WO___9949403A1_I_>

product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism comprising:

a first set of instructions that generate pairwise alignment data for pairs of the polymer sequences, the pairwise alignment data defining regions of similarity between the pairs of polymer sequences with boundaries; and

a second set of instructions that determine additional boundaries in particular polymer sequences by applying at least one boundary from at least one pairwise alignment for one pair of polymer sequences to at least one other pairwise alignment for another pair of polymer sequences including one of the particular polymer sequences, and that generate additional regions of similarity based on the boundaries.

63. The computer program product of claim 62, wherein the computer program mechanism includes:

a third set of instructions that display the polymer sequences with the regions of similarity being spatially aligned with each other.

64. The computer program product of claim 63 wherein the polymer sequences also have regions of dissimilarity, said third set of instructions include instructions that:

horizontally align the regions of similarity and dissimilarity based on the position of each region in the polymer sequence; and

vertically align the regions of similarity among different polymer sequences.

65. The computer program product of claim 62 wherein the second set of instructions includes instructions that:

identify repetitive subsequences within a particular sequence.

66. The computer program product of claim 62 wherein the polymer sequences also have regions of dissimilarity, and the computer program mechanism includes:

a third set of instructions that identify the regions of similarity and dissimilarity, and correlate the regions of similarity and dissimilarity to a biological property.

BNSDOCID: <WO___9949403A1_I_>

67. The computer program product of claim 62 wherein the computer program mechanism includes:

a third set of instructions that identify sequence polymorphisms and correlate the sequence polymorphisms to a biological property.

BNSDOCID: <WO___9949403A1_I_>

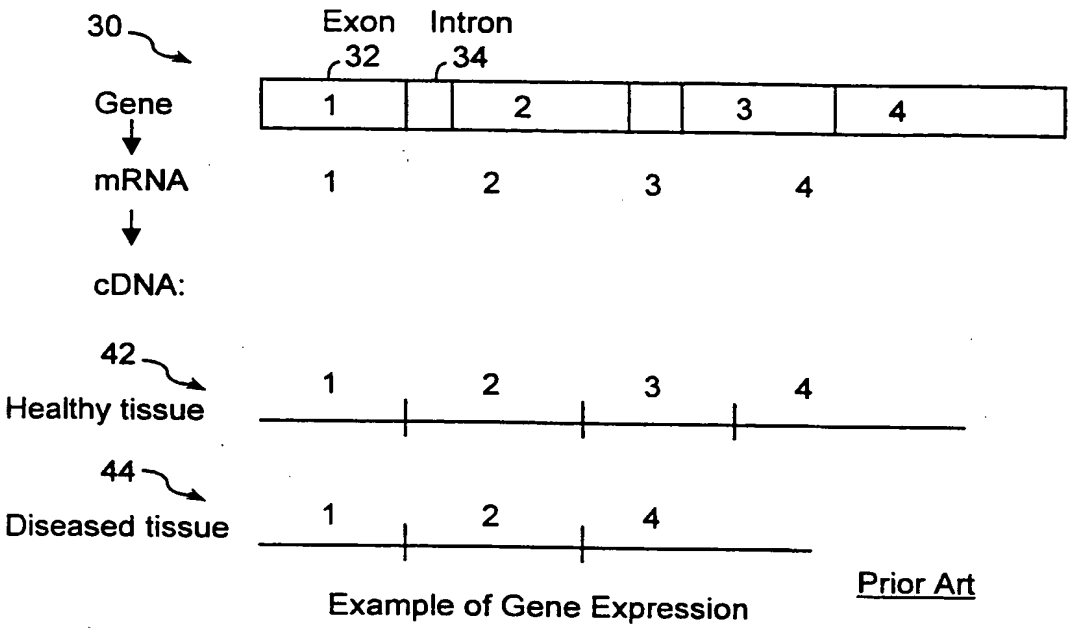


FIG. 1

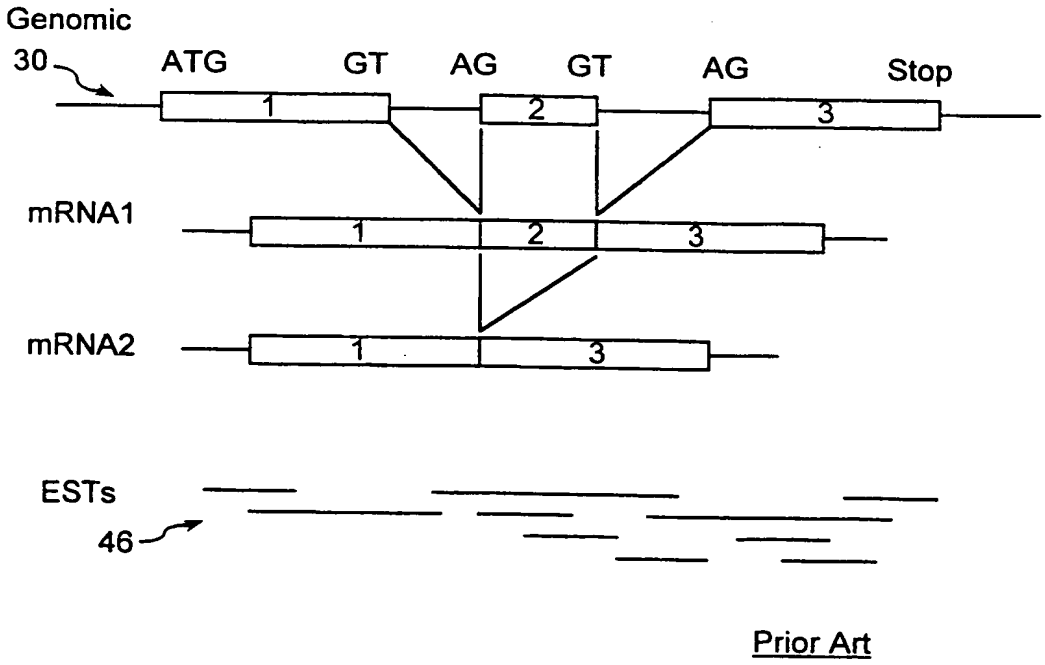
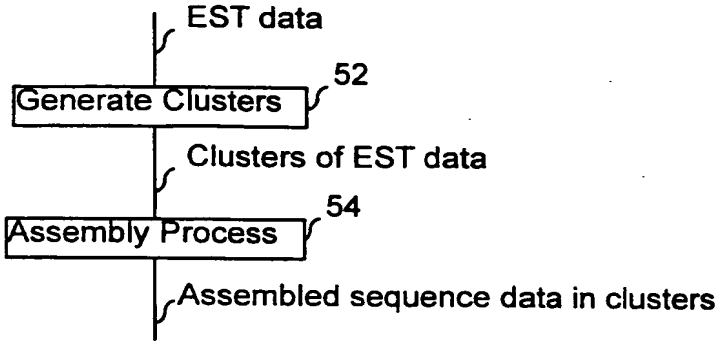


FIG. 2

BNSDOCID: <WO___9949403A1_I_>



Prior Art

FIG. 3

BNSDOCID: <WO___9949403A1_I_>

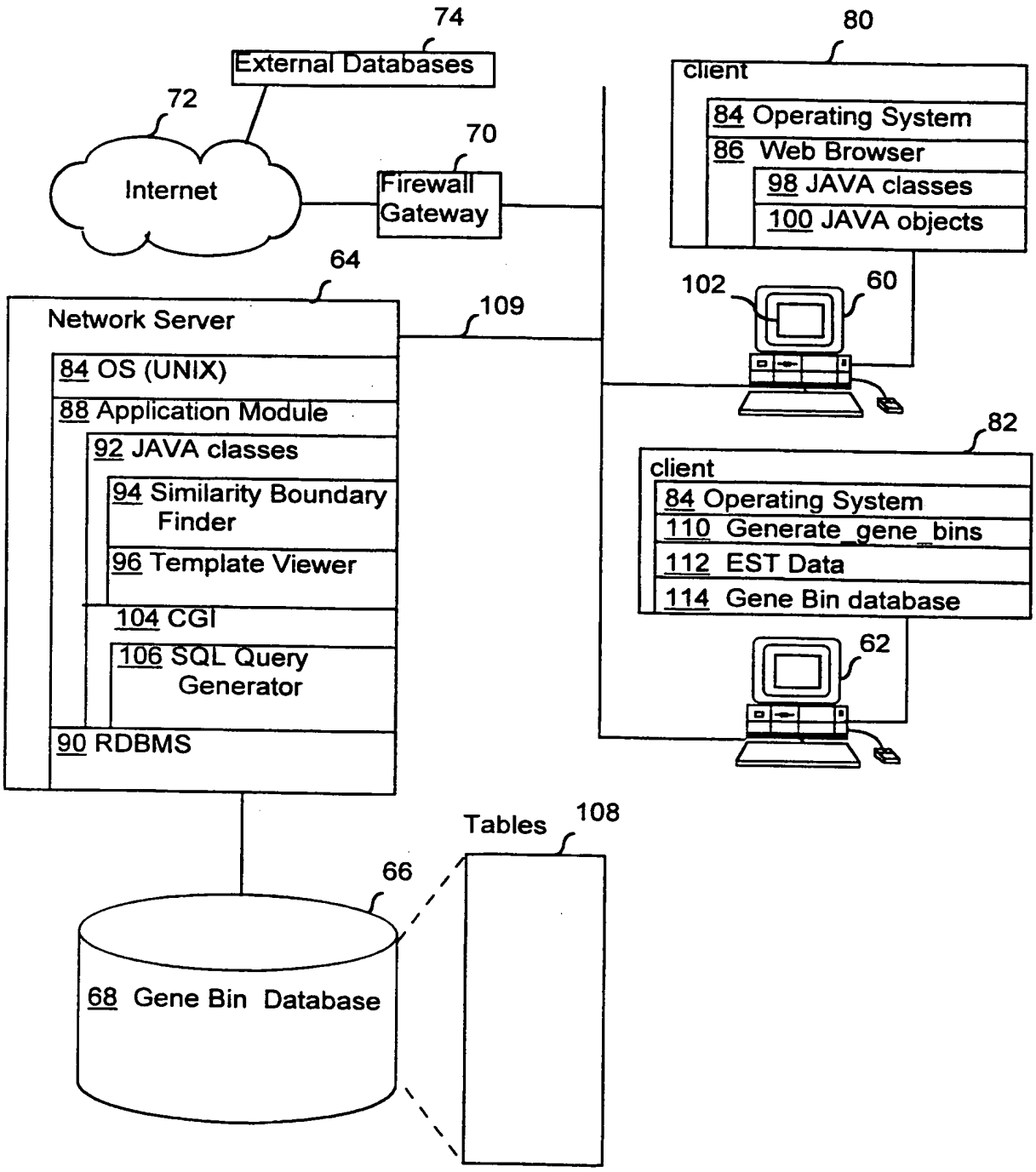


FIG. 4A

BNSDOCID: <WO___9949403A1_I_>

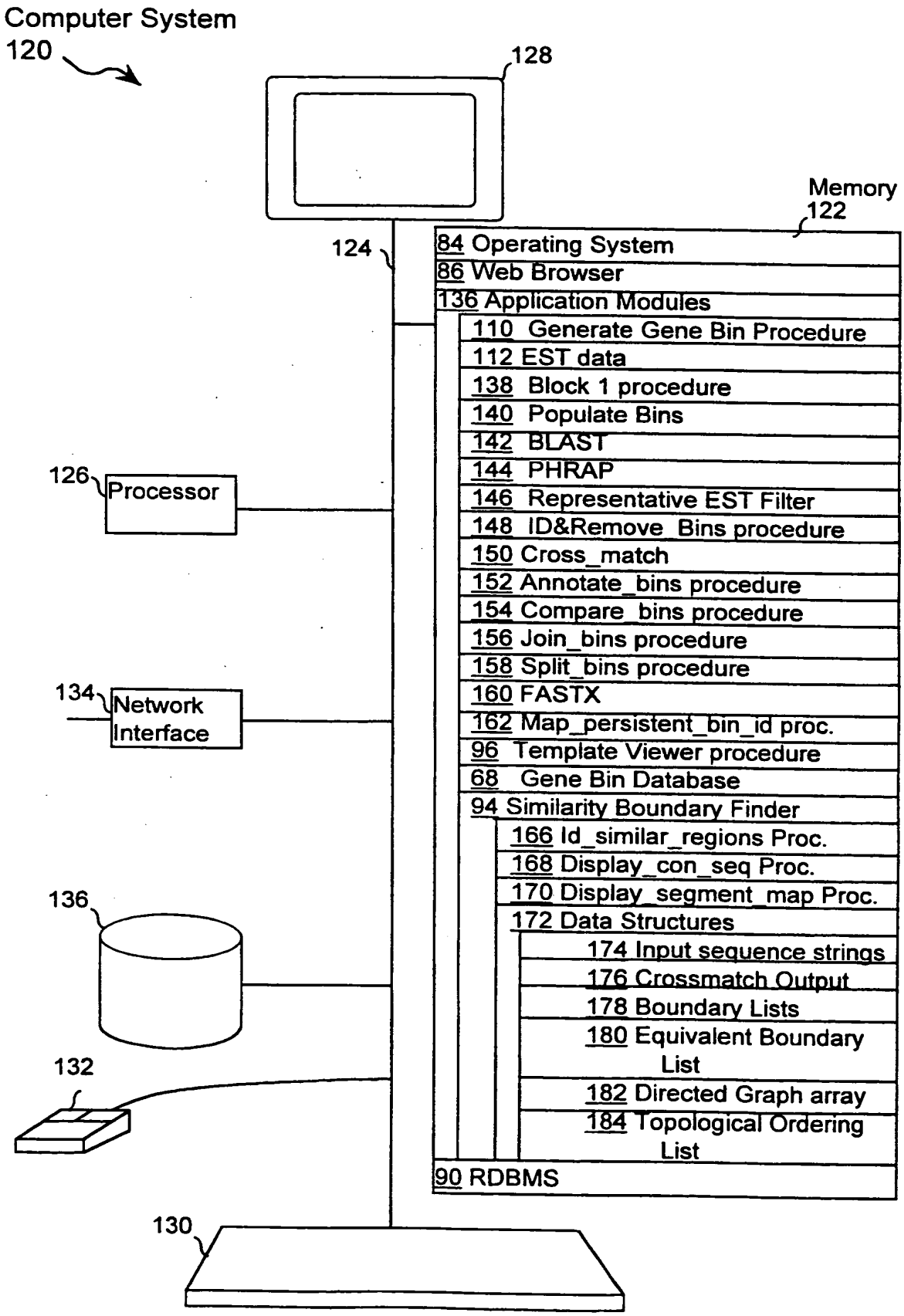
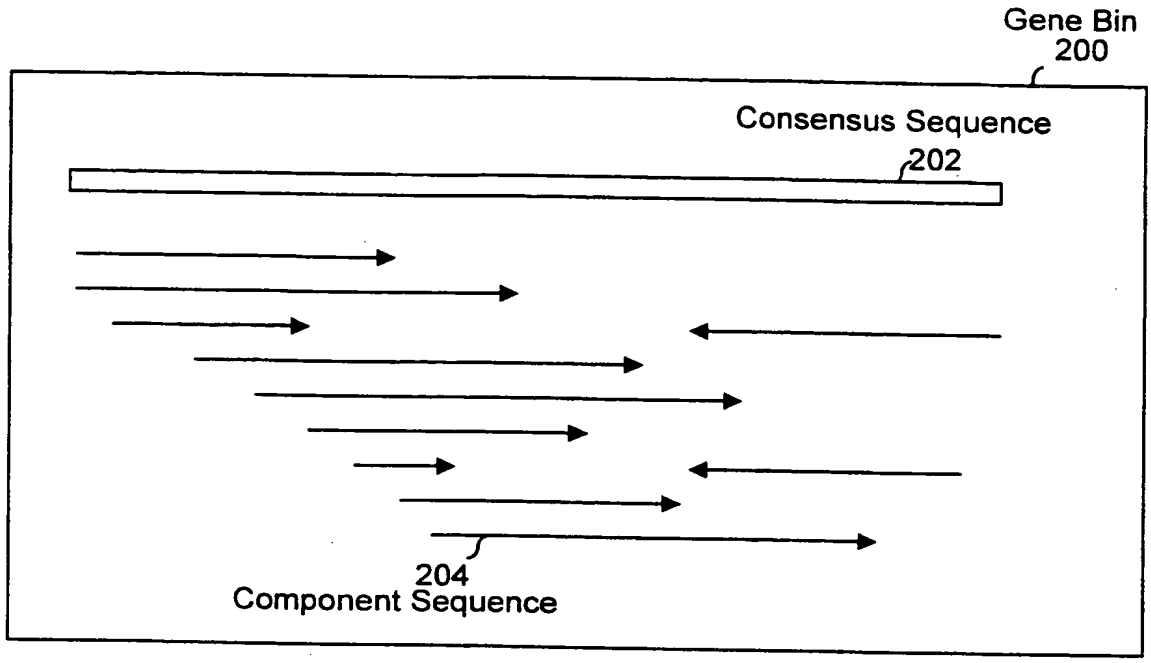


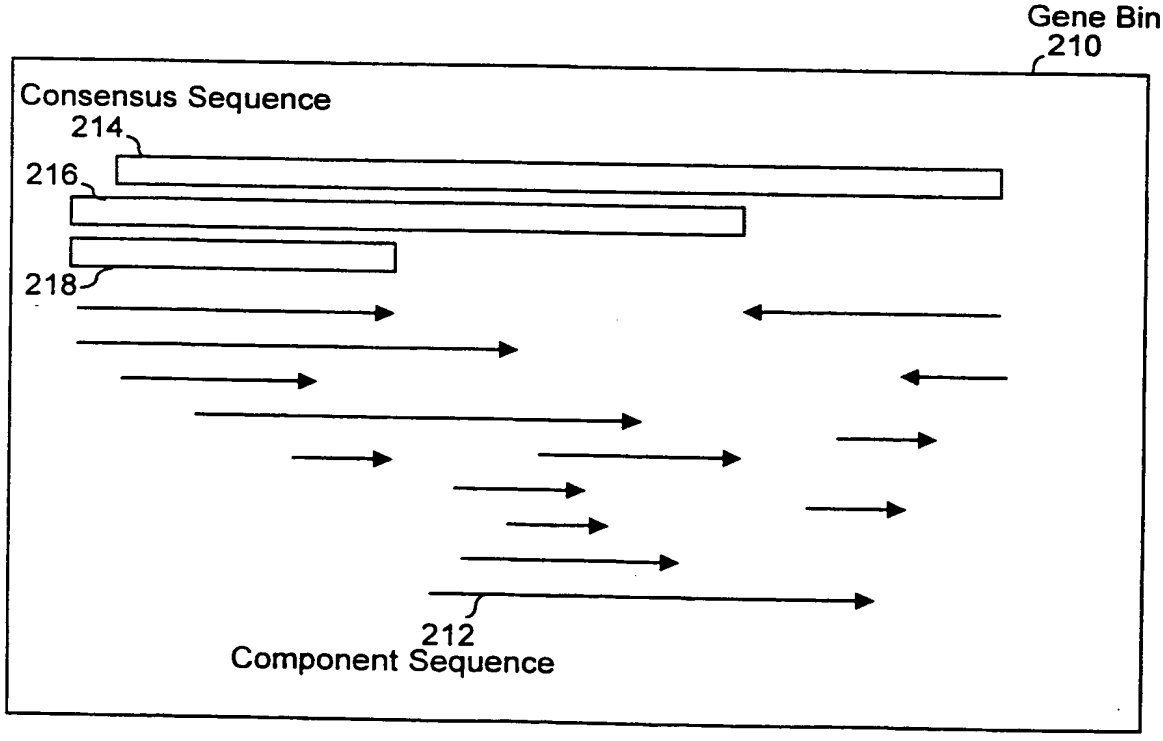
FIG. 4B

BNSDOCID: <WO___9949403A1_I_>



A Gene Bin

FIG. 5A



A Gene Bin

FIG. 5B

BNSDOCID: <WO___9949403A1_I_>

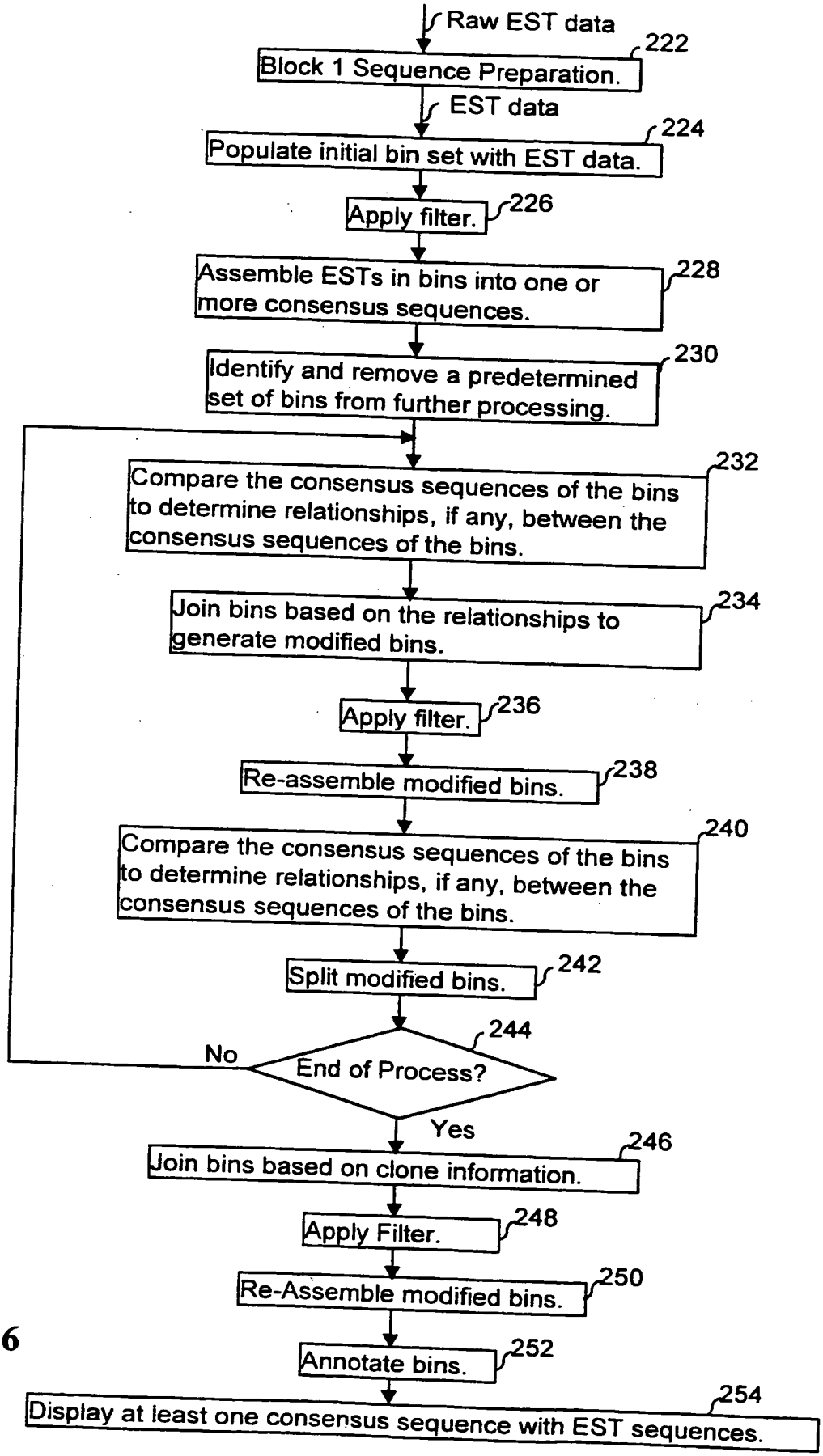


FIG. 6

BNSDOCID: <WO___9949403A1_I_>

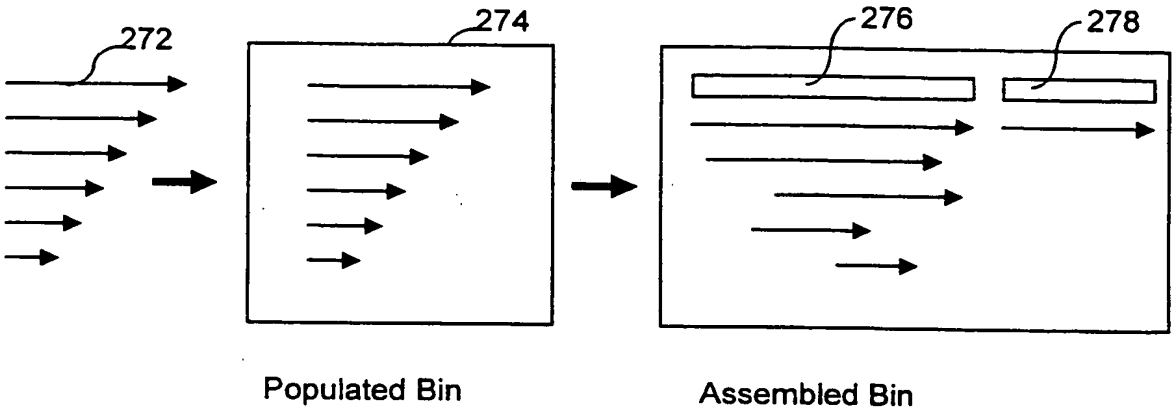


FIG. 7A

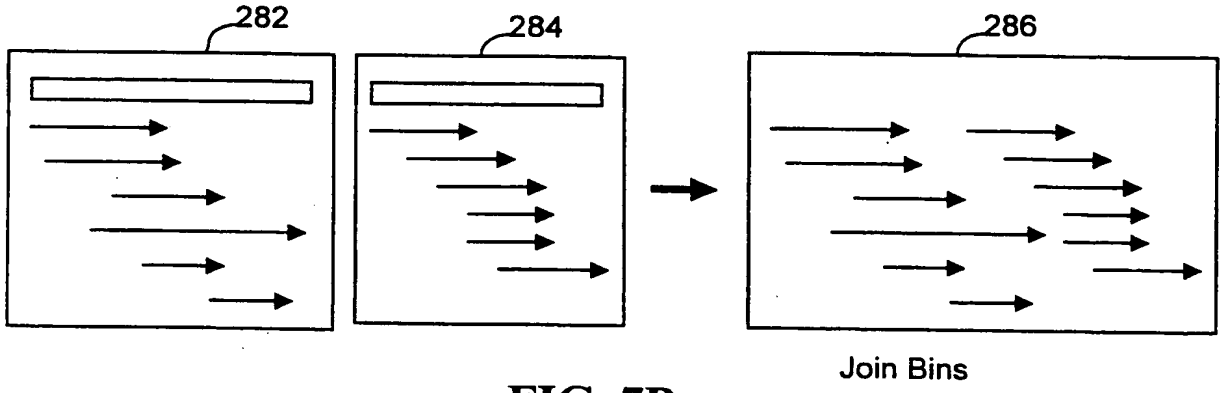


FIG. 7B

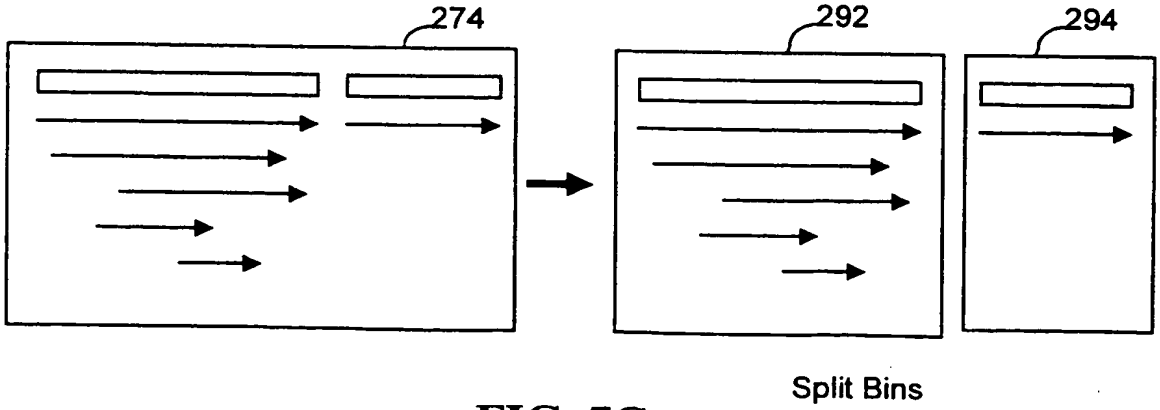


FIG. 7C

BNSDOCID: <WO___9949403A1_I_>

A PHRAP FILTER

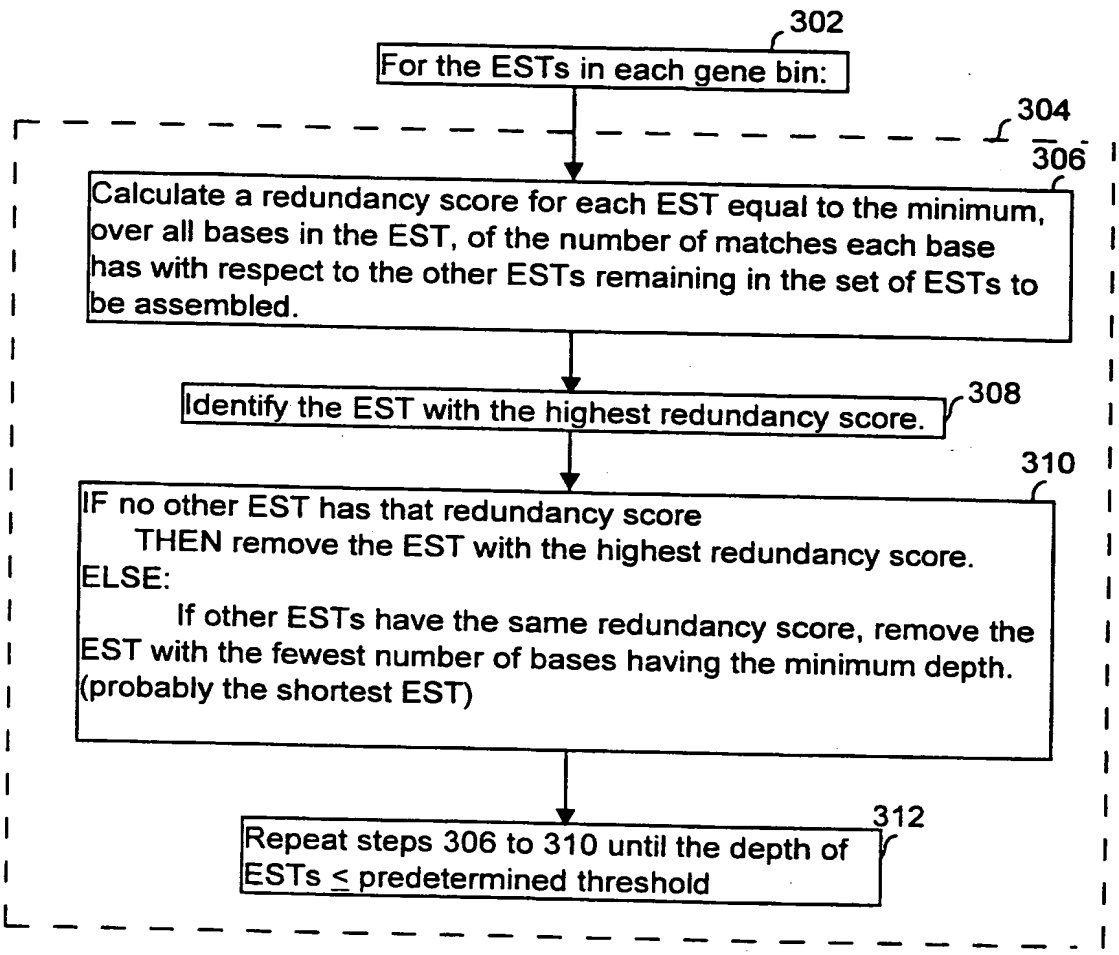


FIG. 8

BNSDOCID: <WO___9949403A1_I_>

MAPPING PERSISTENT BIN IDENTIFIERS

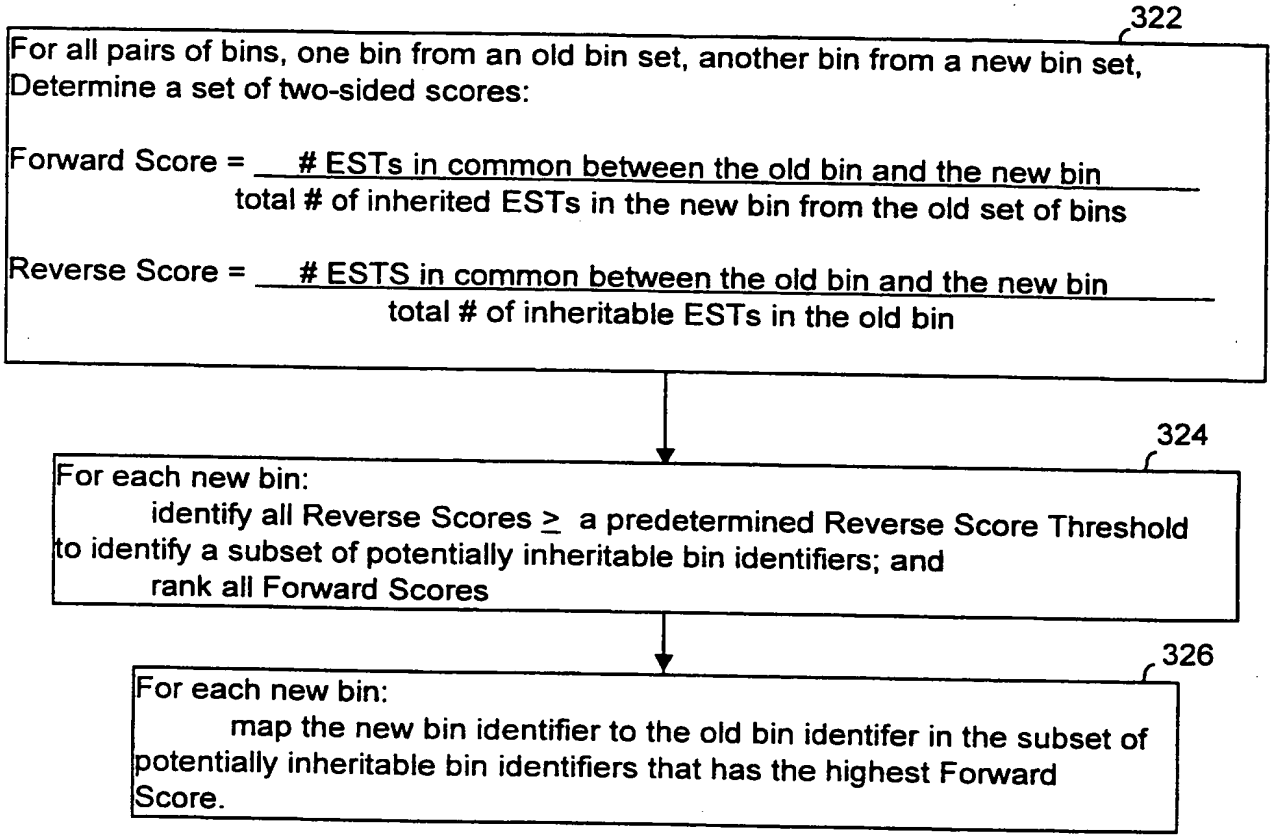


FIG. 9

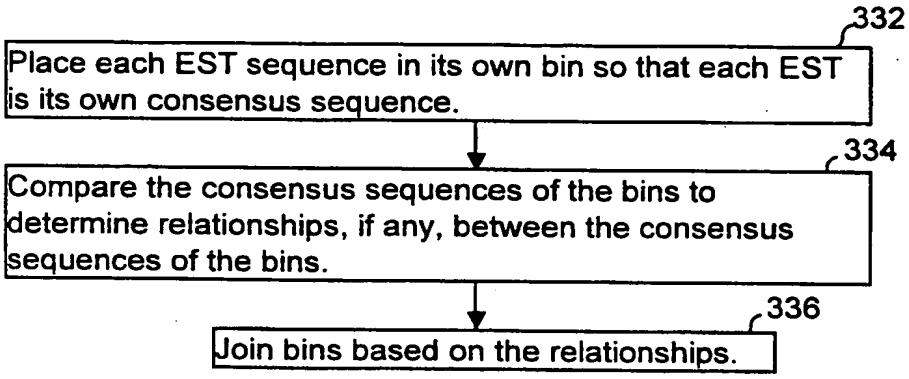
328

Old Bin ID	New Bin ID

Table for Bin Mapping

FIG. 10

BNSDOCID: <WO___9949403A1_I_>



Alternate Method of Populating an Initial Bin Set with EST Data.

FIG. 11

BNSDOCID: <WO___9949403A1_I_>

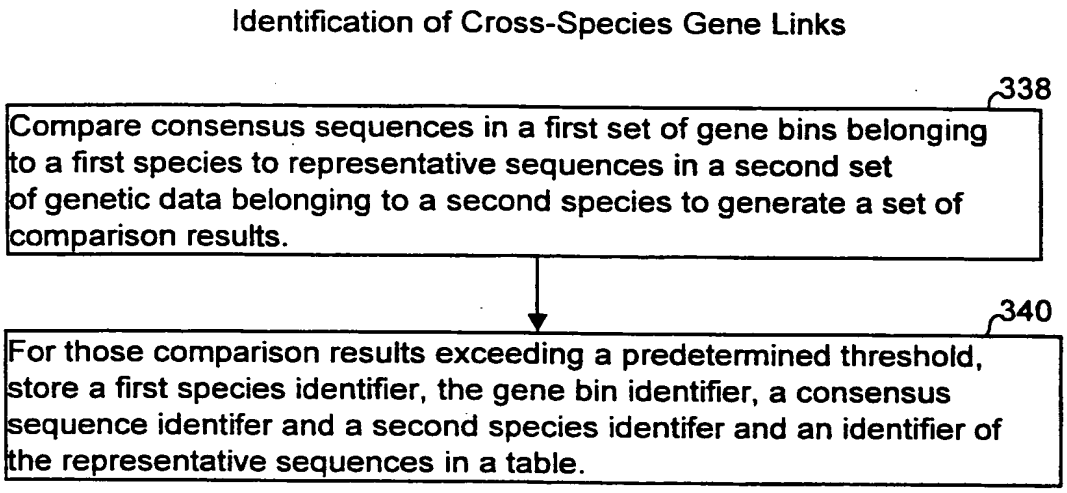


FIG. 12

BNSDOCID: <WO___9949403A1_I_>

GENERAL METHOD OF THE SIMILARITY BOUNDARY FINDER

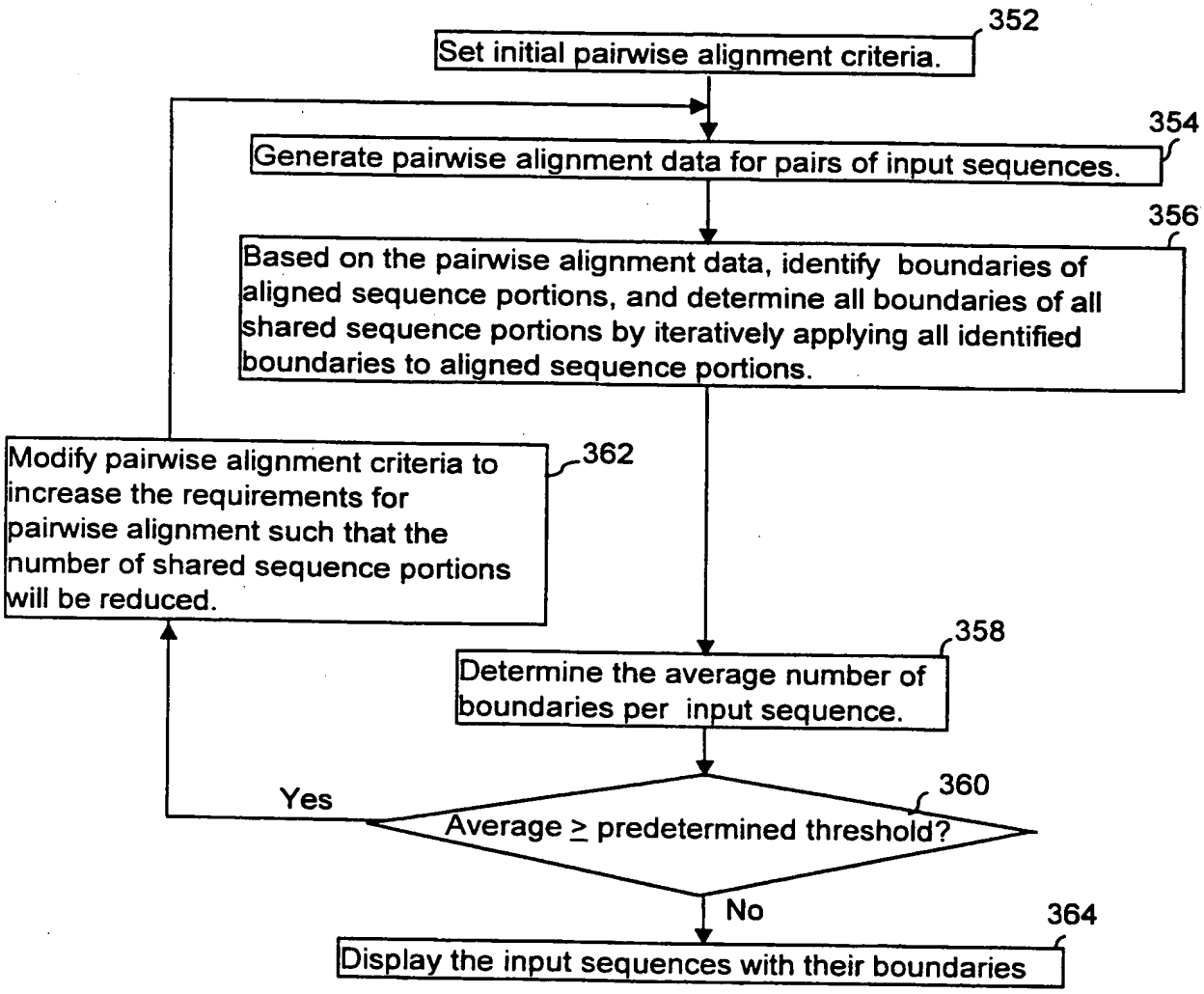


FIG. 13

BNSDOCID: <WO___9949403A1_I_>

ALTERNATE METHOD OF THE SIMILARITY BOUNDARY FINDER

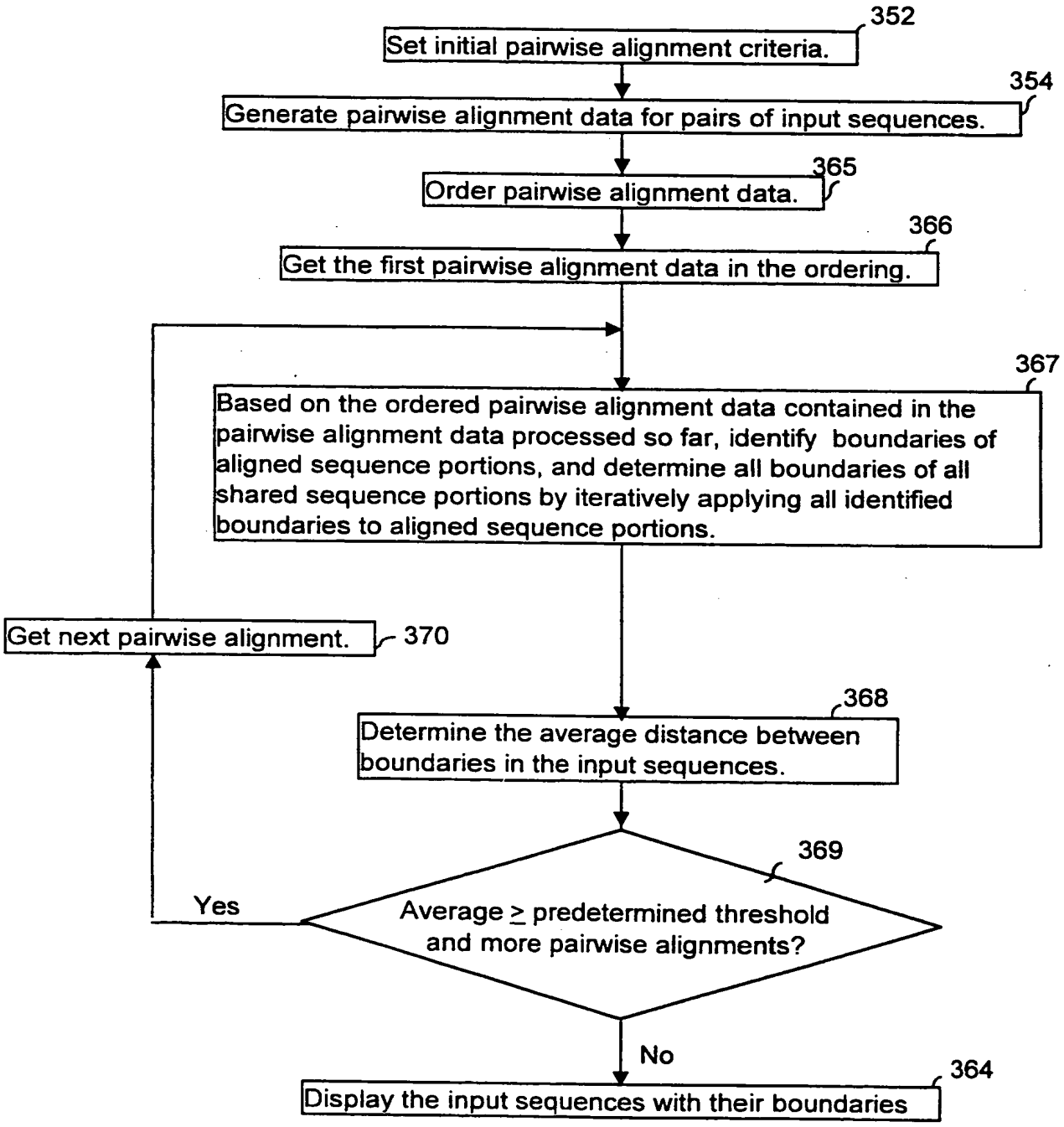


FIG. 14

BNSDOCID: <WO___9949403A1_I_>

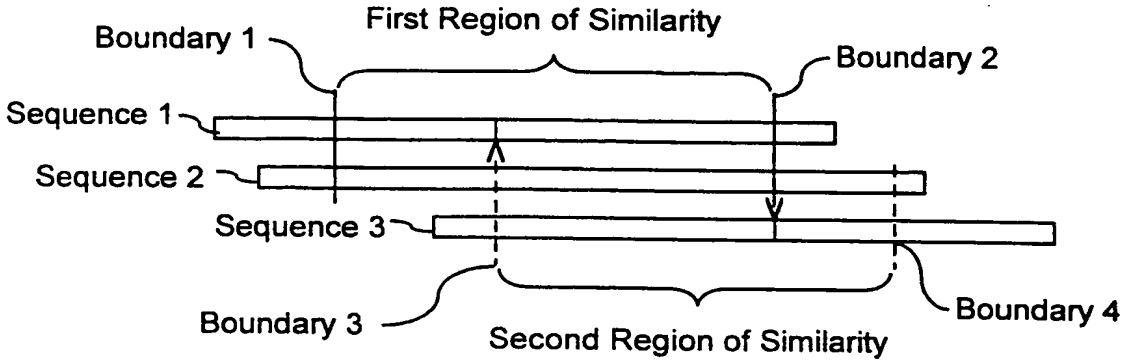

Boundary 1
First Region of Similarity
Boundary 2
Sequence 1
Sequence 2
Sequence 3
Boundary 3
Second Region of Similarity
Boundary 4

FIG. 15

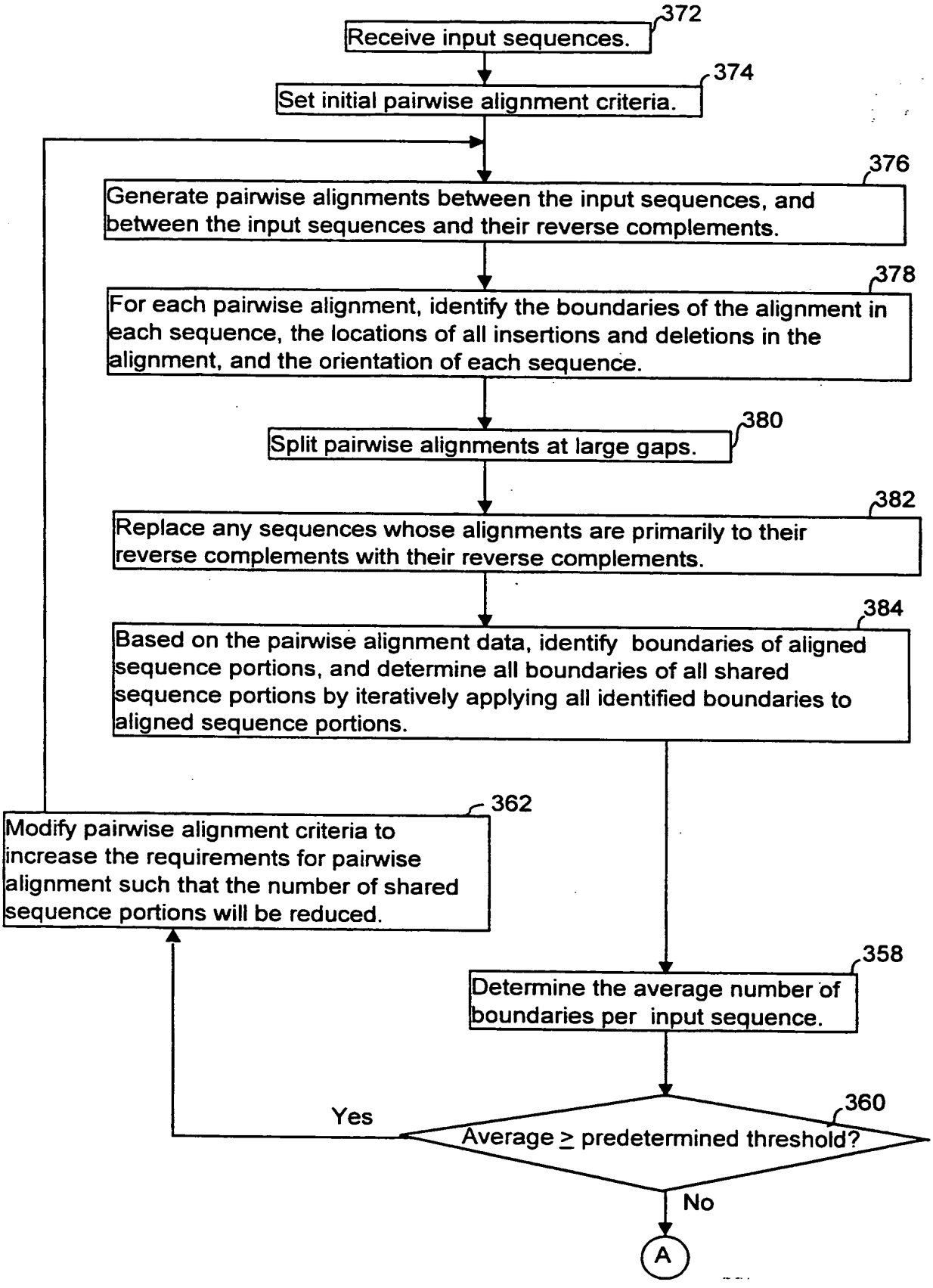


FIG. 16A

BNSDOCID: <WO___9949403A1_I_>

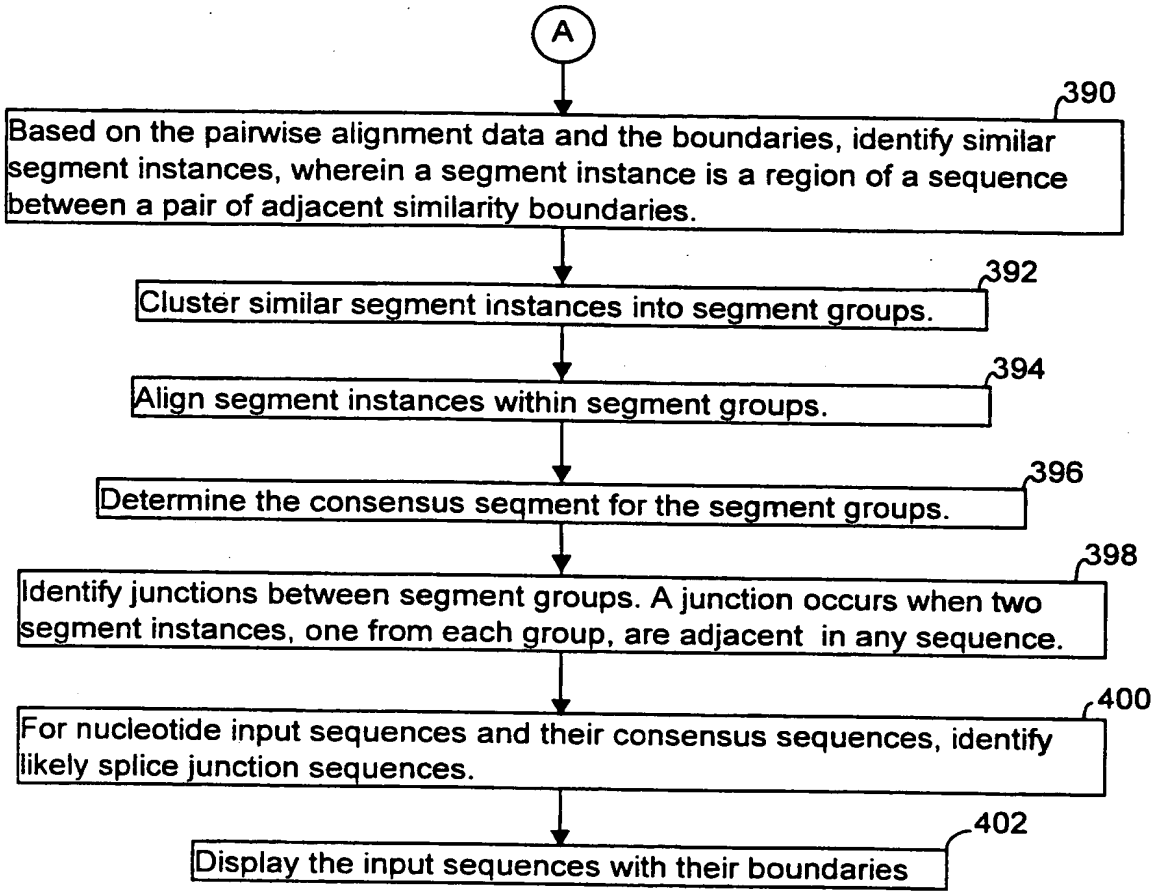


FIG. 16B

BNSDOCID: <WO___9949403A1_I_>

METHOD OF IDENTIFYING AND DETERMINING SEGMENTS WITH MULTIPLE ALIGNMENTS

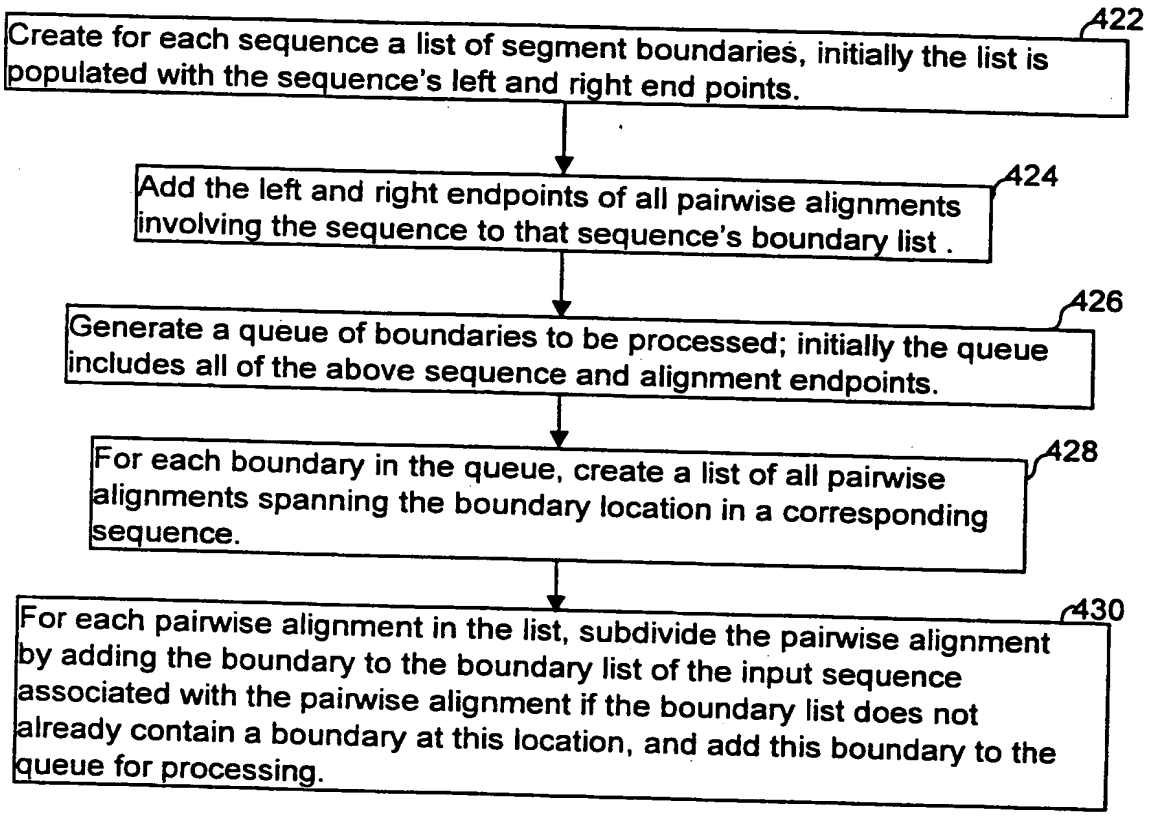


FIG. 17

BNSDOCID: <WO___9949403A1_I_>

EXEMPLARY DATA STRUCTURES

Initial Boundaries from Pairwise Alignment

Sequence Identifer	Sorted Boundary Lists							
Sequence 1	S1	B1	B2					E1
Sequence 2	S2	B1	B3	B2	B4			E2
Sequence 3	S3	B3	B4					E3

Data Structure Storing Equivalent Boundaries

Equivalent Boundaries
Seq. 1 B1, Seq. 2 B1
Seq. 1 B2, Seq. 2 B2
Seq. 2 B3, Seq. 3 B3
Seq. 2 B4, Se1. 3 B4

Boundaries after Subdivision

Sequence Identifer	Sorted Boundary Lists							
Sequence 1	S1	B1	B3	B2				E1
Sequence 2	S2	B1	B3	B2	B4			E2
Sequence 3	S3	B3	B2	B4				E3

FIG. 18

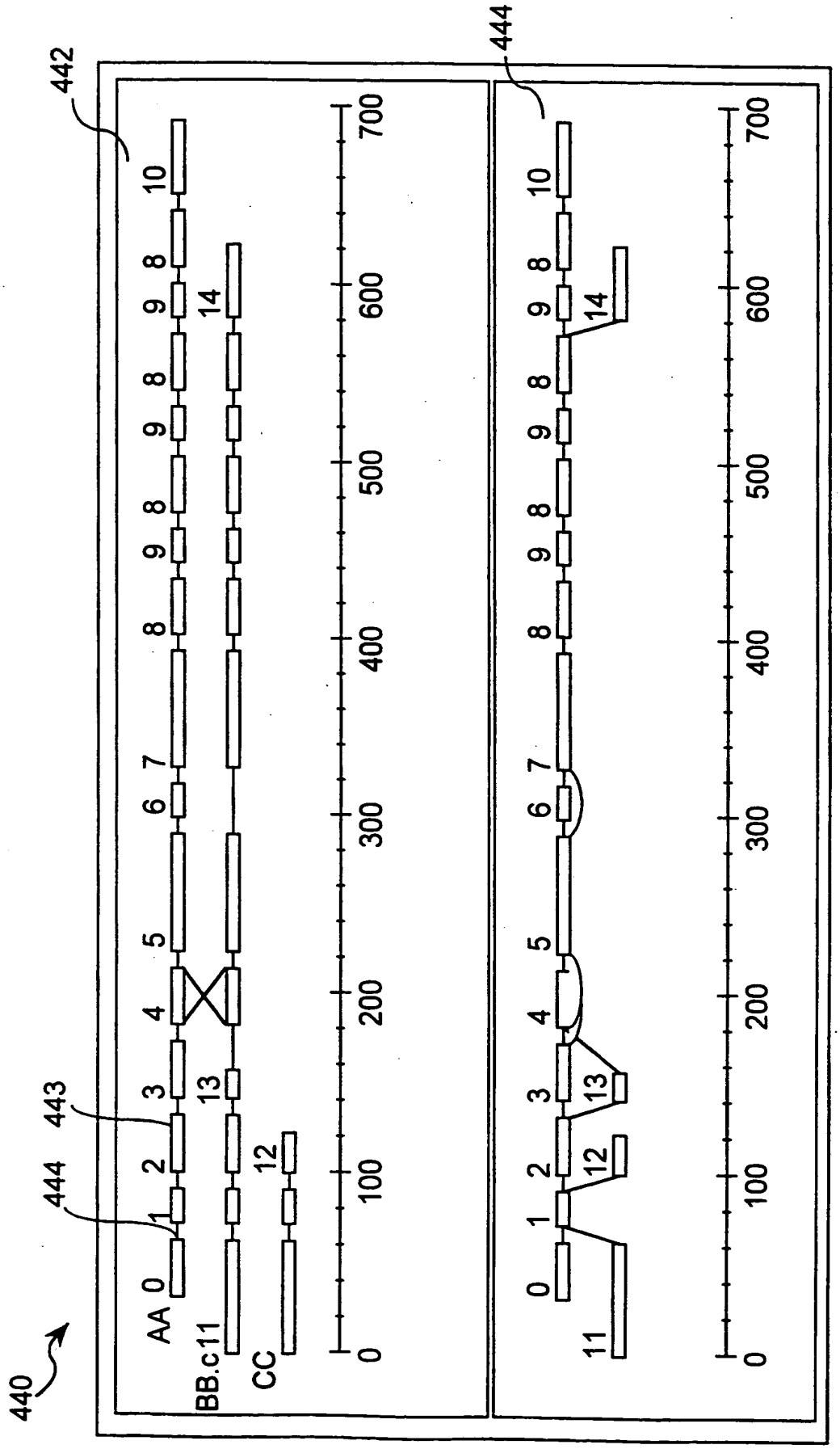


FIG. 19

BNSDOCID: <WO___9949403A1_I_>

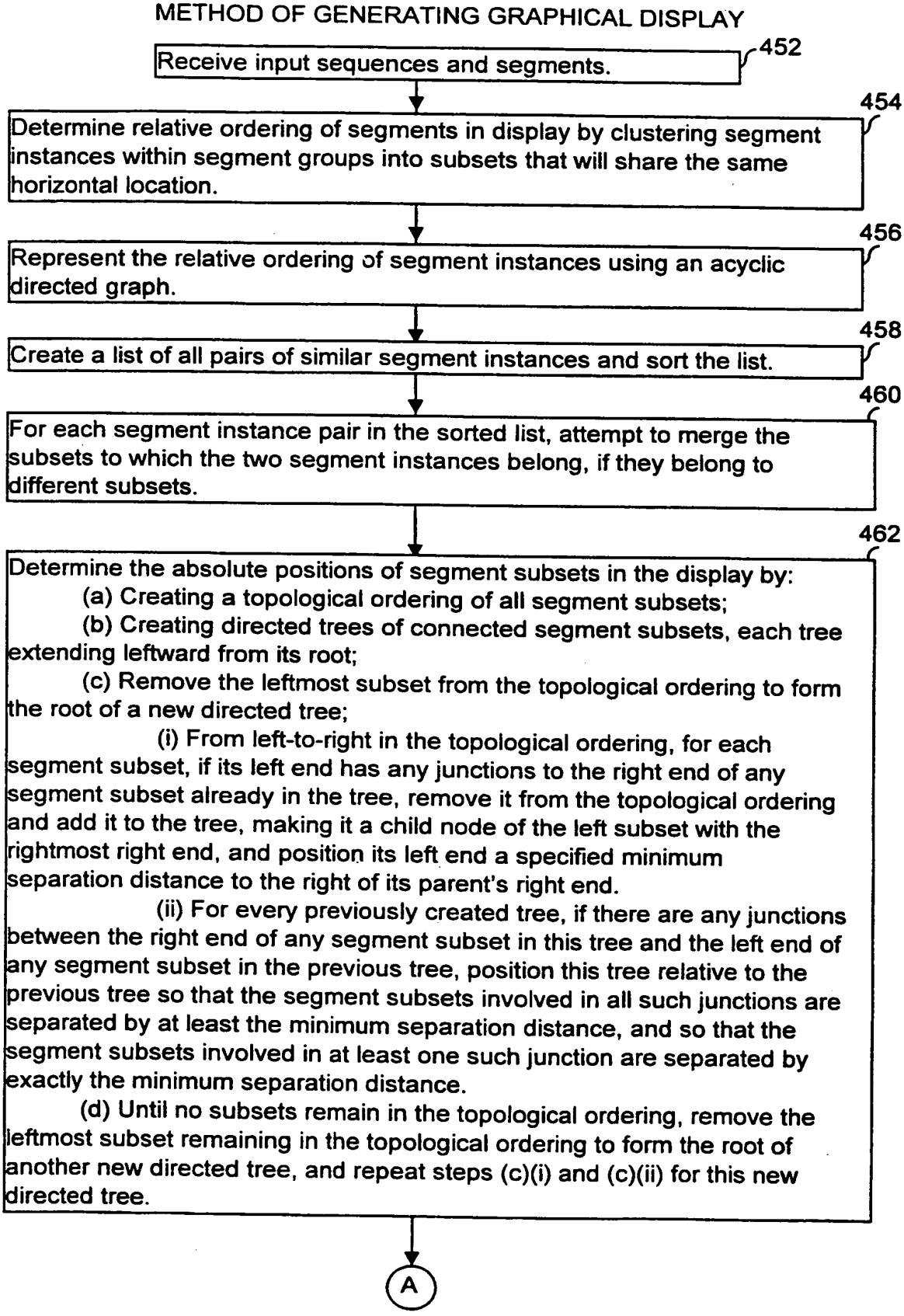


FIG. 20A

BNSDOCID: <WO___9949403A1_I_>

METHOD OF GENERATING GRAPHICAL DISPLAY

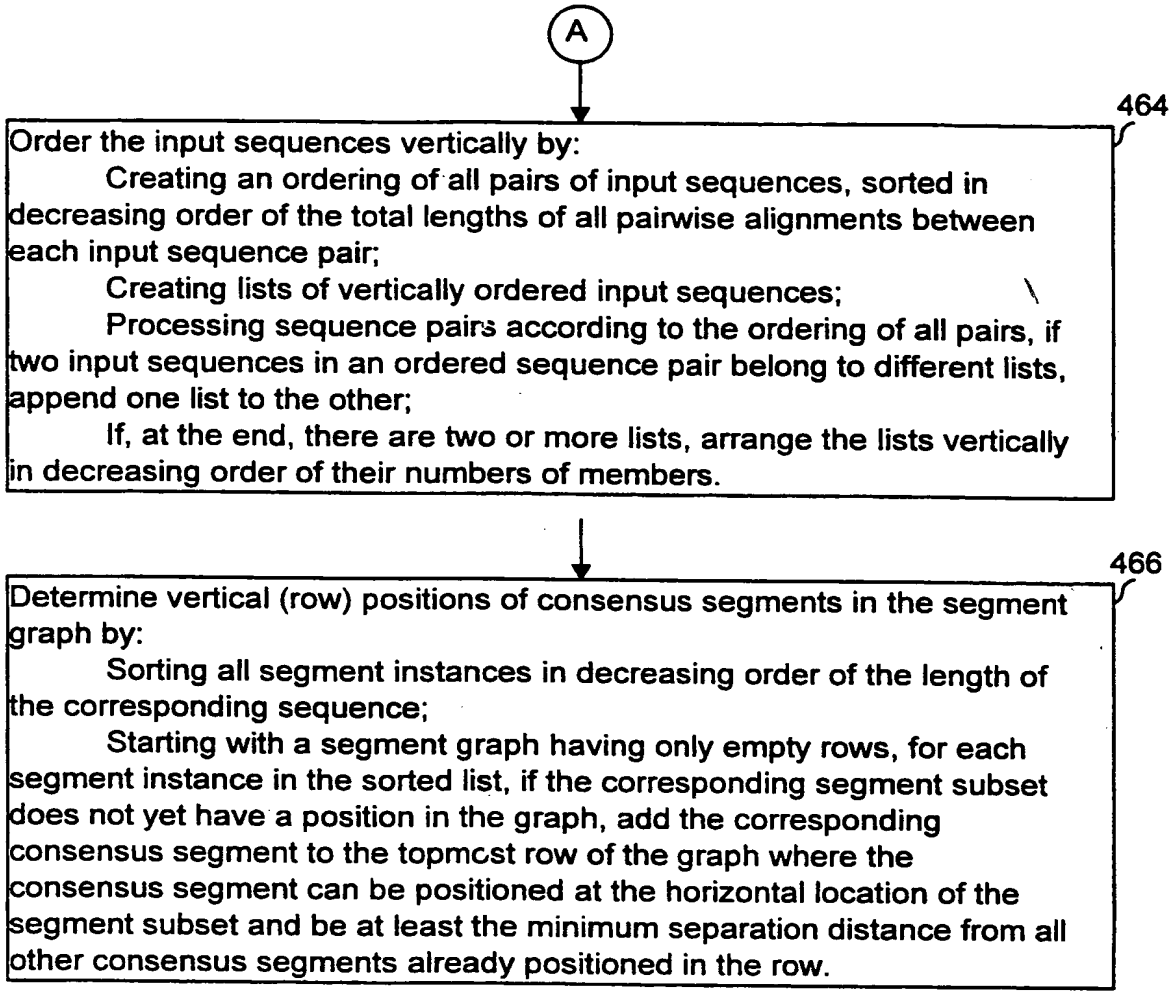


FIG. 20B

BNSDOCID: <WO___9949403A1_I_>